A MODULAR DATA ANALYTIC PIPELINE FOR FEATURE SELECTION IN HIGH DIMENSIONAL MICROBIAL DATA SETS

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
In Partial Fulfillment of the Requirements
For the Degree of Masters
In the Department of Computer Science
University of Saskatchewan
Saskatoon

By

ELLEN REDLICK

PERMISSION TO USE

In presenting this thesis/dissertation in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis/dissertation in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis/dissertation work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis/dissertation or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis/dissertation.

DISCLAIMER

Reference in this thesis/dissertation to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the University of Saskatchewan. The views and opinions of the author expressed herein do not state or reflect those of the University of Saskatchewan, and shall not be used for advertising or product endorsement purposes.

Requests for permission to copy or to make other uses of materials in this thesis/dissertation in whole or part should be addressed to:

Head of the Department of Computer Science
University of Saskatchewan
176 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9 Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9 Canada

ABSTRACT

The demand on the global food supply is ever increasing. With a finite amount of land to grow crops, soil health is crucial to ensuring a continued reliable food supply. Understanding how soil microbiomes affect plant growth has proven difficult in part because of the sheer number of microbes per gram of soil. This challenge is akin to the "large *p*, small *n*" problem in statistics. We have proposed a pipeline to analyze data of this nature with the help of network analysis. Networks, which are commonly referred to in computer science as graphs, are sets of nodes and edges. For the experiments in this thesis, the nodes represent microbes and edges represent their relationships with one another. These relationships are determined by calculating pairwise correlations on the data set. The data used to test the pipeline is an Operational Taxonomic Unit (OTU) abundance table, where columns are OTUs and rows are the samples. Four types of network centralities have been implemented and are used to measure the "importance" of a microbe. Each of these centralities have different interpretations for how to quantify importance.

A sensitivity analysis was performed on a smooth brome invasion dataset using the pipeline. This analysis explored the implications of varying the pipeline parameters, with respect to performance and result consistency. The trade-offs of the parameters are discussed as it is recognized that different users may value different features. This pipeline has been used as part of an application that successfully detected microbes that responded to externalities regardless of abundance.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Kevin Stanley, for his tremendous patience and understanding throughout this project. He taught me the difference between research and industry. He encouraged me to continue when I couldn't see the end. I wouldn't have made it through this master's without his guidance and patience.

I would like to thank Dr. Steven Mamet and Dr. Steven Siciliano for being my gurus for all things soil science.

I want to thank Michelle Brabant for her work on this project as a summer student and for her friendship.

Most of all I would also like to thank my parents, who have encouraged me and given me every opportunity to succeed. I feel incredibly fortunate to have these loving, generous role models in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

OTU	Operational Taxonomic Unit
LSA	Local Similarity Analysis
RMT	Random Matrix Theory
MENA	Molecular Ecological Network Analysis
CoNet	Co-occurrence Network Inference
eLSA	Extended Local Similarity Analysis
MIC	Maximal Information Coefficient
MINE	Maximal Information-Based Nonparametric Exploration
SEM	Structural Equation Modeling

CHAPTER 1

INTRODUCTION

1.1 Motivation

Communities of microorganisms have an unseen but profound impact on everything from human health to agriculture [1]–[7]. Microbial communities comprise 1-3% of the human body mass at approximately 10 microorganisms to 1 human cell [8] and are prevalent at approximately 10^{10} bacterial cells per gram of soil [9]. These communities interact with their substrate and each other in vibrant but usually poorly understood ecological systems. In particular, soil microbial communities are a poorly understood driver of plant growth and phenotype, impacting and impacted by the plants or crops growing in the soil. With global food needs expected to double by 2050 [10], actionable models of how microbial soil communities impact plant growth under changing environmental conditions are critical. Better models of how microbial communities and plants interact in the soil could have a profound impact on the development of soil specific crops, or macrobiotic cultures to enhance or restrict the growth of particular species. However, microbial communities have been historically difficult to analyze, as many soil bacteria and archaea are difficult to isolate, often do not culture well and have uncertain function within their community.

Microbial ecology is the study of microorganisms' interactions and relationships with their environments [11]. When looking at pairwise relationships two species may interact neutrally, positively, or negatively to each other. Common ecological interactions include mutualism, in which both species benefit, parasitism or predation in which one benefits from the loss of the other, and commensalism, where one benefits with no harm or benefit to the other [12]. A common technique is to use mathematical models on microbial abundance data to explain microbial interactions in situ [11]. Advances in metagenomic technologies have, in some ways, made this study of ecological systems more complex and have created questions which require new approaches to answer.

To combat this divergence between the size and scope of the data, researchers have turned to various types of feature selection in an attempt to limit their analysis to a subset of the Operational Taxonomic Units (OTUs) that are considered by some criteria to be important. This feature selection can broadly be binned into two approaches, selection approaches and aggregation

approaches. In aggregation approaches (e.g. [13]–[15]), OTUs are grouped by phylum or function, and analysis proceeds on the groups of OTUs. While this preserves the structure of the communities at a gross scale, it can erroneously group OTUs together, obscuring results. Depending on the level of aggregation, this approach can also lead to overly generalized conclusions which are difficult to replicate or act upon. In the selection approach, OTUs are chosen based on an a priori definition of importance. The most straightforward interpretation of important is using abundance; generally, that more numerous OTUs are more important to the ecosystem [9], [16]. While this approach is simple and intuitive, it can ignore the dynamics of the system, and lead to erroneous conclusions about the relative importance of more marginal species which link populations of OTUs together.

Researchers have also looked at other methods of defining importance, including information-based [17], [18], network-structure based [19]–[21] and approaches based on statistical techniques [12]. Information-based approaches can also be referred to as entropy-based approaches. There are many types of entropy, such as Kullback-Leibler, Quadratic and Shannon, but in general entropy can be used as a proxy for species richness [17].

Network analysis has been used in several ways to learn more about microbial communities. Although several network analysis methods have already been studied, the impact on the results based on the different ways these networks are created has been overlooked. Statistical methods such as regression have been used to predict the abundance of a species based on the abundances of others [12]. Though these methods are often simple to use, it can be difficult to interpret the results in a biological sense and they often cannot model the complicated relationships that exist.

While it is intuitive to assume that different definitions or operationalizations of the relative importance of OTUs would select different sets of OTUs as important, no systemic research has been performed to understand the impact of importance operationalization on OTU selection, or on the selection of parameters and techniques within broad approaches to OTU importance operationalization. The lack of analysis on this front is troubling, as different sets of OTUs downsampled for further analysis would presumably drive different conclusions about the health and nature of microbial communities in soil and elsewhere.

1.2 Research Problem

Metagenomic sequencing creates immense amounts of data per sample. Due to the expense and overhead relating to the process, relatively few samples are generally available. This results in a well-known issue in statistics, referred to as the "large *p*, small *n*" problem. Traditional statistical methods were created based on the assumption of having many observations and few measured features [22], making these techniques inadequate for analyzing metagenomic data. However, ever-increasing computing power opens the door for new methods to be developed.

1.3 Solution

The idea of using graph theory to explore microbial communities has been considered [19], [20], [23]–[27]. We expand upon these concepts, using graph theory and network analysis to determine the important microbes in soil, including rare taxa. In this thesis, we examine the impact of varying the parameters of a novel network-metric based pipeline we created that operationalizes OTU importance. This pipeline was tested on a dataset of a smooth brome invasion in grasslands [28]–[30]. By running a sensitivity analysis, we demonstrate that different OTU operationalizations produce different sets of OTUs, capturing different aspects of the relationship between microbes under changing experimental conditions.

Comparison between the different types of importance definitions exposes two families of microbes: those that are considered important by almost all operationalizations, and those which are considered important for a particular operationalization. This technique opens an intriguing possibility of selecting OTUs using a number of methods to probe those which are important under all circumstances and those which are only important from a specific interpretation of importance.

1.4 Evaluation

We evaluated our solution over two aspects of the problem. First, we determined how consistently individual OTUs appeared in the result set across the different definitions of importance. Secondly, we looked at the stability of the ranked results between all parameter sets using a method called Rank-Biased Overlap. This provided two views into the consistency of results we gathered: the first at a singular OTU level, describing how consistently specific OTUs appeared in the results, and the second on a global scale, describing the stability of the overall set of OTUs that were returned.

1.5 Contribution

This work constitutes a significant contribution to the metagenomics analysis literature for several reasons:

- It identifies the perils of OTU selection based on a single criterion and proposes a simple multi-criteria method to overcome these problems.
- It proposes a solution to metagenomic feature selection in the form of an overall sensitivity analysis and parameter space exploration.
- This work has been proven to be effective in downstream analysis, where OTUs selected as important by the pipeline are used in structural equation modeling to identify keystone taxa [30].

1.6 Thesis Outline

This thesis is organized as follows:

- Chapter 2 consists of a literature review to provide background knowledge required for our research, and a review of related work in the field. We first discuss the development of metagenomic technology, and its impact and limitations with regards to research in soil microbiomes. We then review some of the computer-based approaches that have attempted to solve these limitations.
- Chapter 3 introduces the graph theory concepts used in our solution to the problem. The topics of correlations and graph centralities are discussed, as is the Rank-Biased Overlap method that is required in our evaluation of the pipeline.
- Chapter 4 explains the design architecture of our solution and provides details about its implementation. It also describes the dataset we used and the software libraries that were required.
- Chapter 5 first discusses our sensitivity analysis and defines the parameters that were used. We then evaluate the results from the sensitivity analysis, taking into consideration several factors, including:
    - two measures to evaluate the consistency of results,
    - a review of which pipeline runs returned complete results, and
    - a runtime analysis.

- Chapter 6 concludes this thesis with a discussion and describes an application that has already been based off of this work.

CHAPTER TWO

RELATED LITERATURE

This chapter discusses background information that may be useful for those who are not familiar with soil science and metagenomic sequencing. It also summarizes a number of techniques that have been used to attempt to solve the issues related to our work.

2.1 Metagenomic Sequencing and its Shortcomings

Soil metagenomics is the use of genomic and bioinformatic techniques to explore the soil microbiome [31]. The use of metagenomic sequencing has made analyzing microbial communities and their relationships with their environments a more feasible proposition. Through DNA barcoding using the 16S rRNA or cpn60 sequences in soil samples, operational taxonomic units (OTUs) can be inferred and abundance datasets generated for microbial interaction analysis.

Advances in metagenomic technologies have, in some ways, made the study of ecological systems more complex and have created questions which require new approaches to answer. 16S rRNA PCR amplicon sequencing is useful to achieve the common goal of identifying the diversity of a microbial environment because the 16S gene is found in every organism and has a slow evolutionary rate, which allows identification of characterized genera and potential classification of novel ones [32].

Derived from the work of Carl Woese in the late 1970s [33] small subunit 16S rRNA sequencing has become a well-established means of identifying the taxonomy and phylogeny of microbes. Using the 16S small subunit ribosomal RNA gene is valuable for a number of reasons: it is found in prokaryotic organisms; material extracted from dead cells is unlikely to be included as viable genes, as RNA cannot survive for long once the organism is no longer living [34]; and it has highly conserved regions, allowing for the creation of universal primers [32], [35].

Another way to explore microbial diversity is through sequencing of the chaperonin 60 (cpn60) coding region for which universal target primers have been developed. This type 1 chaperonin sequence, discovered by Hemmingseng et al. in 1988, is present in both prokaryotes and eukaryotes, allowing for a more comprehensive look at microbial diversity with the inclusion of fungal taxonomies [36]–[38]. A curated database of eukaryotic, bacterial, and archaeal cpn60

sequences has been developed to exploit the ubiquity of cpn60 for microbial diversity analyses [36].

The use of metagenomics sequencing has made assaying microbial communities a more feasible proposition. However, despite the advances high throughput sequencing has induced in metagenomics, metagenomic analysis has several well-documented shortcomings including sequence assembly, taxonomic classification, and chimeras [35], [39], [40].

When performing 16S-based sequencing classification, sequences with fewer than ten reads are commonly found to be noise [41]. To prevent these from being mistakenly classified as OTUs, it is common practice to exclude sequences with fewer than 10 reads that are not found in replicate sequencing [40].

Chimeras are a type of error that comes from sequencing when multiple parent sequences are incorrectly merged together to identify a new taxon that does not actually exist, falsely inflating the perceived community diversity. Because chimeras will typically be classified as low abundance OTUs, the correct detection and removal of chimeras becomes important when considering rare taxa. Both false negatives and false positives will alter the relative abundances of the community. Because of this, several bioinformatics tools have been developed to try to mitigate chimeras [35].

Another issue with metagenomic sequencing is, possibly counterintuitively, the amount of data generated. This stems from the fact that the estimated microbial diversity in one gram of soil could exceed the catalogue of known prokaryotes [42]. However, metagenomic soil analysis is an expensive and time-consuming process, therefore only dozens of samples are assayed per experiment [43].

Sequencing technologies have revealed the vast diversity of soil microbiomes, and led to the discovery of a plethora of previously unknown rare species known as the rare biosphere [34]. As the discovery of new microorganisms through sequencing techniques increases, so does interest in this rare biosphere [44]–[46]. Relatively little is known about the influence of these microbes, leading to interest in finding new methods to measure importance of all microbes in the biome.

Depending on the criteria for labeling, typical metagenomic soil assays can generate hundreds or even thousands of candidate OTUs. When trying to measure the role OTUs play in an ecosystem, a traditional method is to look primarily at abundance, with the notion that the more

abundant the OTU is, the more important it is [47]–[49]. As one can surmise, this method is highly insufficient when considering the possible importance of the species in the rare biosphere. Although there are still some issues in sequencing microorganisms in soil environments, being able to analyze the large amounts of data these sequencing methods produce aside from their abundance is a major hurdle that needs to be addressed. With measurements in the dozens and variables in the thousands, traditional statistical analysis cannot provide meaningful insight from the data, as the statistical power of any direct comparisons would be essentially nil.

Although there are still some issues in sequencing microorganisms in soil environments, being able to analyze the large amounts of data this sequencing produces is a major hurdle that needs to be addressed. As the discovery of new microorganisms through sequencing techniques increases, so does the interest in the rare biosphere [44]–[46]. When trying to measure the role OTUs play in an ecosystem, a traditional method is to look primarily at abundance, with the notion that the more abundant the OTU is, the more important it is [47]–[49]. A significant amount of evidence has been found that some of these rare species play important roles ecologically [6], [34], [50]–[54]. Therefore, we can infer that basing importance on abundance is highly insufficient when analyzing ecosystems and that new methods must be created to properly asses importance.

2.2 Existing Approaches to Analyzing High Dimensional Data sets

2.2.1 Aggregation techniques

One straightforward approach when analyzing the exorbitant number of OTUs found in microbiomes is to aggregate OTUs to a higher level on the phylogenetic tree [13], [14], [55]. One reason for aggregating OTUs at a higher level, such as the genus level, is that it makes the analysis more robust to sequencing errors [55]. As sequencing technology gets more advanced, this concern is becoming less valid. Shi *et al.* [14] used this aggregation approach to determine if there is an association between body mass index (BMI) and gut microbiome. They did so by running their analysis on OTUs grouped at a genus level and then again on OTUs grouped into four sub-compositions of each genus. By doing the sub-composition analysis, they were able to find a genus that was associated with BMI after adjusting for fat and caloric intakes. During the preprocessing

of this study, after the OTUs were grouped into genera, those that had zero counts in more than 90% of the samples were removed from the analysis. This is potentially problematic as the analysis will omit the impact rare species may have. Similarly, grouping the OTUs generalizes the results, where there may be a single OTU that is associated with BMI that could otherwise be further isolated.

Analyzing rare features in high-dimensional data sets is a problematic task in several disciplines. In addition to the microbiome datasets we are focused on, it commonly occurs in text mining, where algorithms are used to interpret the sentiment of bodies of text based on certain terms appearing in the work. In both domains, aggregation techniques are often employed to handle rare features. Aggregation strategies were used to determine hotel ratings based on written reviews while taking into account rare words [15]. They implemented an aggregation operator that was effective in grouping similar, rare adjectives with the help of a tree of existing knowledge from past datasets. They also demonstrated that ordinary least squares and lasso methods were ineffective when rare features were included without aggregation. However, because this method results in a grouping of features, we lose the ability to select individual rare features that may be more significant than others.

2.2.2 Approaches to deal with rare species

One argument for keeping rare species is that they might be a case of under-sampling. For example, if the total count of a sample is small, then the probability of an observed zero being due to under-sample increases dramatically. Therefore, rare species that are found in these samples are more likely true positives [56].

It can be concluded that dealing with rare species is a problematic topic with differing opinions on best practices. The significant evidence that has been discovered about the importance of rare species would imply that including them in analysis is prudent, though it is still important to remove erroneous ‘species’ that appear due to sequencing errors.

2.2.3 Hybrid approaches

Multiple approaches have been used in analysis to increase the robustness of the solution [3], [5], [26]. These ensemble methods recognize the strengths and weaknesses of each individual method and purport that combining them will find the optimum solution. Especially from a biological

perspective, different types of functional relationships may be better identified by different types of network creation [57].

2.2.4 Existing methods using network analysis

The concept of using graph theory to further explore microbiome ecosystems has been considered with regards to selecting significant features in a large data set [19], [20], [23]–[27]. There are many different methods that have been used to create these networks. Correlation networks have been commonly used in ecological network inference. These types of networks use a correlation method to create the edges (relationships), using pair-wise correlations, between the nodes (microbes). These edges may represent biologically meaningful relationships between two microbes [57]. For example, if two microbes benefit from each other in the microbiome, it can be assumed that they will be positively correlated [57]. It was established that microbes can also be indirectly affected because a tendency for positive correlation was discovered in phylogenetically related microbes [3].

Researchers have created these correlation networks using many different methods. Common statistical methods like the Pearson, Spearman, and Kendall Tau correlation coefficients have been regularly used, but newer methods like Maximal Information Coefficient (MIC) and Local Similarity Analysis (LSA) have been proven more effective at identifying more complex relationships [50], [58], [59].

One system that expands on creating networks based on pair-wise correlations is a Random Matrix Theory-Based Network approach [20]. Random Matrix Theory (RMT) was first proposed by Wigner and Dyson in the 1960s and has traditionally been used in physics, though it has recently been used with success when applied to biological systems as well [19], [20], [60], [61]. Molecular Ecological Network Analysis (MENA) is a two-part process with the first step relying on an RMT-based approach [20]. After the abundance data is standardized, a pairwise correlation of the data was performed. RMT is then applied to determine the adjacency matrix to use to create an undirected graph. The second part of the process is analyzing the graph using network analysis methods such as network topology characterization. The main proposed benefit of this system is that the RMT method is able to automatically identify a threshold for the network construction.

A project called Co-occurrence Network inference (CoNet) was developed to determine co-occurrence patterns in abundance data [5], [57], [62]. It uses several correlation measures at

once in an ensemble method. These measures are compared with each other and an edge is created when the measures agree that an interaction exists. This approach aims to predict relationships that are more complicated than what is captured in simple pairwise relationships. The argument against pairwise relationships is that they cannot describe complex interactions where a taxon influences, or is influenced by, several others. This research was later expanded into the CoNet app, which is a plugin developed for Cytoscape [62] for easy visualization and operation. The researchers noted several pitfalls in the network creation and inference, including normalization issues and biases introduced during the processing of samples.

Local Similarity Analysis (LSA) is useful when building association networks in a temporal setting [57], [58], [63]. A local similarity score is computed between two sequences that have been standardized with a normal score transformation [58], [64]. This idea is similar to approaches that have been used for local alignment of DNA sequences [58]. It was able to find co-variance relationships that may have been missed by other approaches that do not support time series data. However, the permutation procedure used to calculate the local similarity has been found too computationally intensive, leading researchers to look for performance improvements on this idea [65].

Extended Local Similarity Analysis (eLSA) was created based on traditional LSA to provide faster analysis of time-series data without losing accuracy [65]. Performance improvements are highly advantageous when analyzing microbiome data as the size of the datasets is often prohibitively large.

2.3 Summary

In this chapter, we reviewed metagenomic sequencing and explained its associated shortcomings. We then reviewed several existing approaches to analyzing high dimensional data sets, focussing on network creation and analysis methods. Although we concentrated on research relating to OTU selection, we also touched on an application in natural language processing. We reviewed why it is important to find selection methods that include rare species. We intend to expand upon existing concepts, using graph theory and network analysis to determine the important microbes in soil, including rare taxa.

CHAPTER THREE

BACKGROUND

The techniques described in this thesis rely heavily on mathematical constructs which may not be familiar to many readers. A brief overview of these constructs is presented here for clarity.

3.1 Introduction to Graphs

In the feature selection scenarios considered in this thesis, the response of a microbial community to an experimental manipulation (natural or designed) is envisioned. We wish to identify microbes and microbial communities which in some sense respond together to the experimental manipulation. While statistical techniques are adept at identifying pairwise mutual variation (this is essentially the definition of correlation), these techniques are less adept at identifying how these changes diffuse through the community. To model the network of change dependencies other mathematical constructs are required. One construct commonly used in fields from computer networking to the epidemiology of contagious disease is the network diagram, or more formally a graph, as pictured below in figure 3.1.

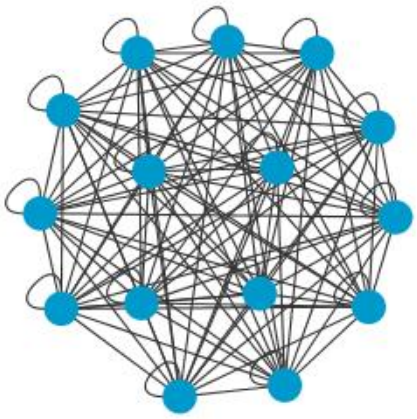

Figure 3.1 – A visual representation of a graph. The nodes (microbes) are represented in blue, and the edges (relationships between two microbes) are shown in black.

In the broadest sense, a graph is a mathematical object with points and connections between them [66]. More specifically, graphs are networks consisting of vertices (nodes) and edges (arcs) [67]. A simple definition can be given as:

$$G = (V, E) \tag{3.1}$$

where V is a set of vertices and E is a set of edges.

An edge connects two vertices and may have an associated weight. Weighting of an edge is typically used to describe the importance of the connection between the two vertices, and can be derived many ways [68]. A graph is described as connected if for every two vertices, there is a path, often passing through intermediate vertices, between them. When this condition is not met, the graph is described as disconnected. This becomes important when analyzing the relationship between items (vertices) represented by a graph.

Due to the complex nature of biological systems, some researchers have turned to graph theory to help explain the connections in these networks [17], [21], [23], [25], [68]–[70].

3.1.1 Correlation Types

Because we are interested in how OTUs respond to an experimental manipulation, building graphs based on established notions of correlation or correspondence is a necessary first step. From a biological perspective, it has been suggested that a positive correlation between two microbes may imply a mutually beneficial interaction [23]. Conversely, negative correlations could imply competition between microbes or a predator-prey relationship [23].

In order to build these networks, we need to find the pairwise relationships between the features of a large dataset. Though there are several correlation techniques available, we looked at two different types of correlation: Spearman Rank and the Maximal Information Coefficient (MIC). These methods were chosen because of their use in past studies [21], [24], [50], [57], [59], [69].

3.1.2 Spearman Rank Correlation

The Spearman Rank Correlation is a nonparametric correlation test that measures the strength and direction of association between two ranked variables [71]. This is one of the most common statistical correlations and has been used in many ecological studies [21], [24], [57], [69].

3.1.3 Maximal Information Coefficient

MIC is part of the set of maximal information-based nonparametric exploration (MINE) statistics that are used to identify and classify relationships between two variables [59]. MIC is able to ascertain associations that are both functional and non-functional. To calculate the MIC on a set of two-variable data, the data is partitioned into different grids for each grid resolution (x,y). For each of these x-by-y grids, the mutual information $m_{x,y}$ is calculated, and the maximum score for each resolution is normalized to a value between 0 and 1. A character matrix $M = (m_{x,y})$ is defined, which holds the highest mutual information score for any of the x-by-y grids. Then the MIC is the maximum value in the character matrix M. MIC is then the maximum the value in M. This process is illustrated in Figure 3.2 below.

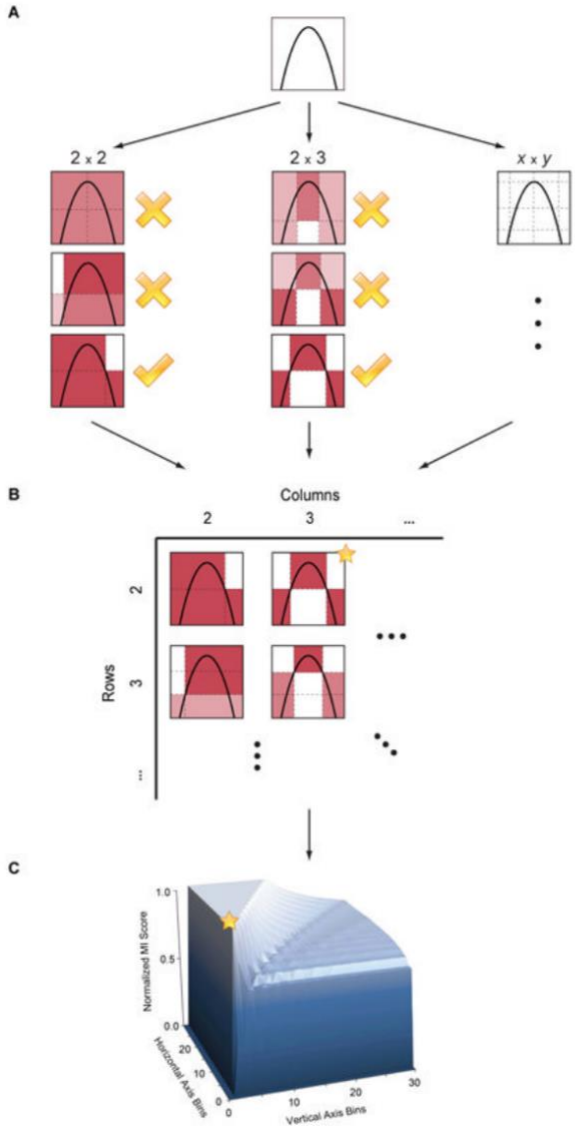

Figure 3.2 - The Maximal Information Coefficient is calculated by exploring the mutual information score of several grid resolutions and layouts of grids within those resolutions. For each grid layout in a resolution, the mutual information score is computed for each grid structure (a). The scores are normalized and the highest value for each resolution is stored in a matrix (b). Those scores are visualized as a surface, where the highest point on the surface represents the MIC. Image from D. N. Reshef et al [59]

Reshef defines the MIC formally as: "For a grid G, let I_G denote the mutual information of the probability distribution induced on the boxes of G, where the probability of a box is proportional to the number of data points falling inside the box. The (x,y)-*th* entry $m_{x,y}$ of the characteristic matrix equals $max\{I_G\}/log\ min\{x,y\}$, where the maximum is taken over all x-by-y grids G. MIC is the maximum of $m_{x,y}$ over ordered pairs (x,y) such that $xy < B$, where B is a function of sample size; we usually set $B = n^{0.6}$". [59]

This technique of partitioning the data into an optimal grid is able to consistently detect results that are not dependent on the relationship type.

3.2 Centrality Types

Graphs can be represented as matrices, where a connection between two vertices is represented by either a 1 or 0 in the case of unweighted graphs or a value, typically between 0 and 1, for weighted graphs. However, these matrices do not directly quantify the concept of connectedness, as only the immediate neighbors of each vertex are encoded. To better operationalize connectedness, network centralities can be employed. Though there are many types of centralities that can be evaluated, for the purposes of this project we look at four centralities: degree, closeness, betweenness, and eigenvector. These centrality types are used to describe how important a node is in the network, based on different interpretations of important.

Centralities encode values of importance for how nodes relate to each other in a network, and have accompanying biological interpretations, as described in the following sections.

3.2.1 Degree Centrality

Degree centrality describes how many immediate neighbors a node has. Nodes with a large number of direct connections are known as hubs. Because these hubs are so highly connected, their removal will have a substantial impact on the topological features of the network. The calculation of degree centrality for a node is often normalized, resulting in the following definition:

$$Degree(i) = \frac{\sum_{j \neq i} a_{ij}}{N - 1} \tag{3.2}$$

where a_{ij} represents an edge between nodes *i* and *j*, and N is the number of nodes in the graph.

The graph in Figure 3.3 shows the degree centrality for each node. Node D has the highest degree centrality value because it has the most direct edges.

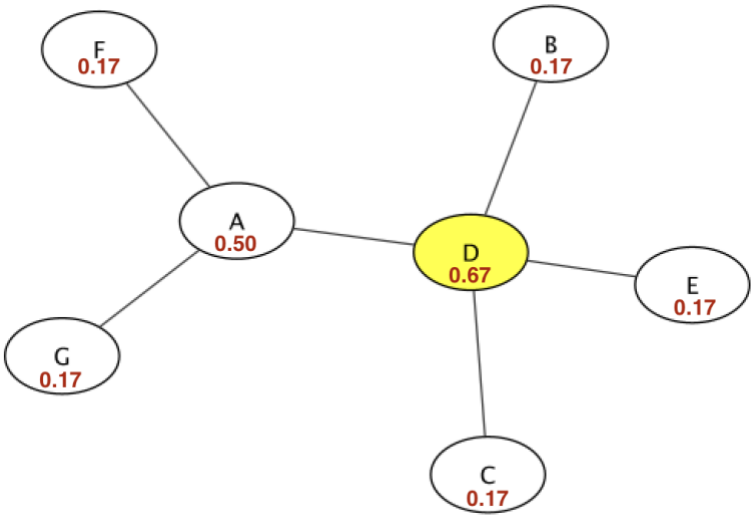

Figure 3.3 - Degree centralities calculated on nodes in a graph

From a biological prospective, several studies have shown that the removal of hubs causes major disturbances in the functionality of the network, known as the centrality-lethality rule [68], [72]. Jeong used degree centrality to analyze a protein interaction network [73], and determined that the degree could identify cell proteins that were most vital to its survival. From a biological perspective, degree centrality is simply the "total number of shared connections with other taxa" [30].

3.2.2 Closeness Centrality

Closeness centrality is defined as the reciprocal of the sum of the number of edges needed to get from the node to all other nodes. The closeness of a node indicates how quickly the node can communicate with the other nodes in the network. More precisely, closeness of a vertex *i* can be defined as:

$$Closeness(i) = \frac{1}{\sum_j dist(i,j)} \tag{3.3}$$

where *dist*(*i*, *j*) is the distance of the shortest path between the nodes *i* and *j* [68].

The graph in Figure 3.4 displays the closeness centrality values for each node. Node D has the highest closeness centrality value because it requires the fewest connections for this node to reach all other nodes.

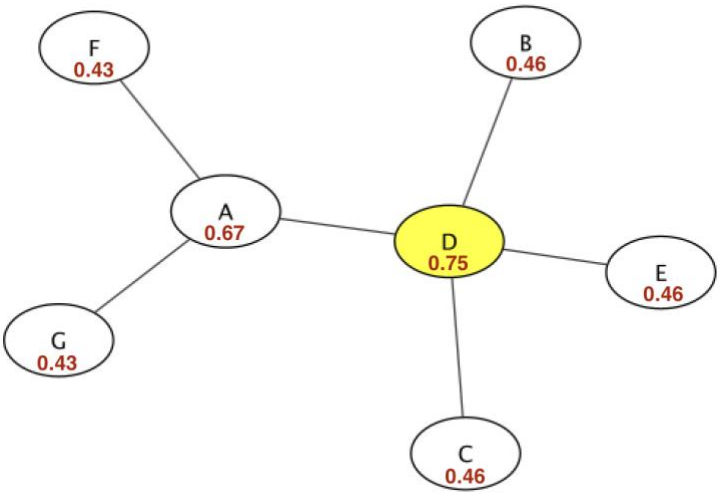

Figure 3.4 - Closeness centralities calculated on nodes in a graph

A node with a high closeness shares many connections with other nodes and is therefore more central to the network. This type of centrality has been used in multiple studies to analyze metabolic networks, determining the metabolites were the most central to the network. This allows researchers to identify which metabolites can be changed into other metabolites with the fewest number of steps [74], [75]. The da Silva study also looked at betweenness centrality and degree centrality but found that closeness was the most appropriate centrality for this application. Another study looking at keyphrase extraction by graph centrality measures found that closeness was very effective on small dataset, but the least effective on large datasets [76].

In a biological network, closeness centrality identifies taxa that are close to many other taxa, based on some measure of association [30].

3.2.3 Betweenness Centrality

Betweenness centrality is a measure of how many times a node is part of the shortest path between two other nodes. As such, betweenness can identify nodes that are important in the interactions between nodes. A formal definition of betweenness can be defined as follows. Take distinct nodes $i, j, w \in V(G)$ with σ_{ij} as the total number of shortest paths between i and j. Then $\sigma_{ij(w)}$ is the

number of shortest paths from *i* to *j* that pass through *w*. For $w \in V$, let $V(i)$ be the set of all ordered pairs, (i, j) in $V(G)$ x $V(G)$ such that *i, j, w* are distinct. We can then calculate betweenness as [68]:

$$Betweenness(w) = \sum_{(i,j)\epsilon V(w)} \frac{\sigma_{ij}(w)}{\sigma_{ij}} \tag{3.4}$$

The graph in Figure 3.5 shows the betweenness values for each node. Node D has the highest betweenness centrality value because it is on the shortest path between the all other pairs of nodes the most often.

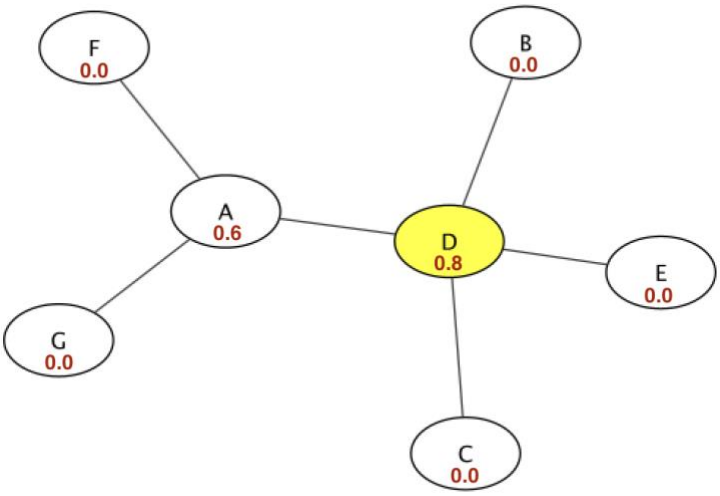

Figure 3.5 – Betweenness centralities calculated on nodes in a graph.

Relating back to metabolic networks, the betweenness centrality describes how often a metabolite participates in a metabolic conversion [75]. With regards to proteins, a high betweenness indicates that they are "key connector proteins with essential functional and dynamic properties" [77].

When applied to OTU covariance graphs, "taxa with high betweenness are those that share connections between modules that do not share many intra-module connections, representing a potential pathway for resource sharing between modules" [30].

3.2.4 Eigenvector Centrality

Eigenvector centrality considers a node to be more important if it is connected to other nodes that are important to the network. To calculate eigenvector centrality, "each vertex i is assigned a weight $x_i > 0$, which is defined to be proportional to the sum of the weights of all vertices that point to i: $x_i = \lambda^{-1} \sum_j A_{ij} x_j$ for some $\lambda > 0$, or in matrix form $Ax = \lambda x$, where A is the (asymmetric) adjacency matrix of the graph, whose elements are A_{ij}, and x is the vector whose elements are the x_i." [78]

Figure 3.6 shows the eigenvector centrality values for each node in the graph. Node D has the highest eigenvector data because it is considered the most influential in the network. This is due to the number direct connections it has, and the number of connections those connections have.

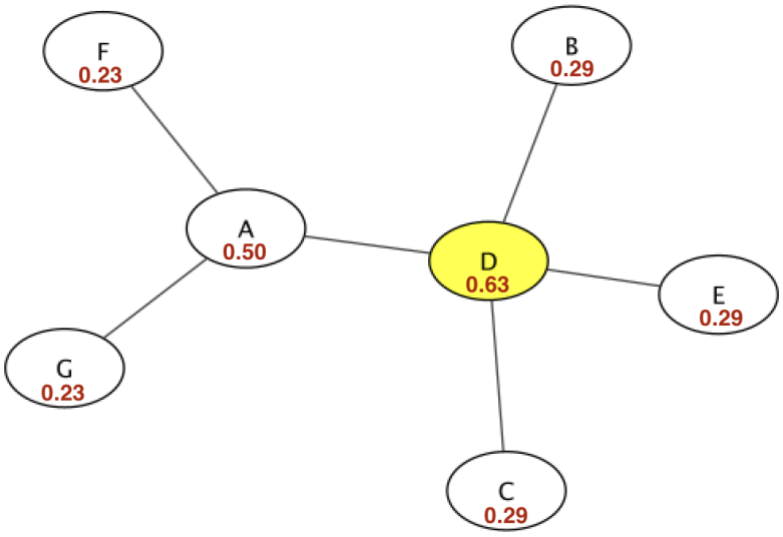

Figure 3.6 - Eigenvector centralities calculated on nodes in a graph.

Due to the complexity of solving large equations, the power method is often used to compute the largest eigenvalue of the adjacency matrix. The power iteration method numerically estimates the value of the largest eigenvalue. This method is resolves quickly when there is a dominant eigenvalue. When there is no dominant eigenvector, the number of iterations needed can increase to a point that hinders performance. Therefore, this method is often suspended after a

specified number of iterations. If the power iteration method does not converge on an answer in that number of iterations, no answer will be returned.

Eigenvector centrality is most commonly known for its use as a page ranking system found on the internet [79]. It has also been used in several biological studies, including identifying gene-disease associations [80], evaluating hubs in yeast protein interactions [72], and measuring the effects of brain stimulation of adults of different ages [81].

In recent years, eigenvector centrality was used to find a relationship between the makeup of the soil microbial community and above ground plant health in tobacco [19]. When applied to OTU covariance eigenvector centrality "measures the tendency for a focal taxon to share connections with other taxa that have connections with many other taxa" [30].

3.3 Rank-Biased Overlap

Rank-biased Overlap (RBO) is a measure of similarity between ranked lists of items. RBO can be used for non-conjoint lists. It is considered an indefinite ranking, meaning it satisfies "the qualities of top-weightedness, incompleteness, and indefiniteness" [82]. Top-weightedness means that items higher in a list should be considered more important than those further down the list. Incompleteness refers to the case where lists do not represent the full rankings of their domains. Some elements are in the set will be in one ranking list but not in the other. Indefiniteness means that the similarity can be calculated independent of the depth of the ranking list [83].

The value of the RBO score is in the range [0,1]. A score of 0 will exist when the lists are disjoint; that is, no element in list 1 is contained in list 2 or vice versa. A score of 1 indicates identical lists.

Formally, Webber defines rank-biased overlap on infinite lists as [82]:

$$RBO(S,T,p) = (1 - \mathrm{p}) \sum_{d=1}^{\infty} p^{d-1} \cdot A_d \tag{3.5}$$

In this formula, *S* and *T* are two ranked lists. *d* is the depth or position in the list, and A_d is the proportion of overlap at depth *d*. *p* is a user-set parameter [0,1] indicating the strength of the top weighting in the calculations. The smaller the *p* value, the more top-weighted the calculation will be. A *p* value of 0 will result in only the top-ranked item being considered.

3.4 Summary

This chapter introduces key concepts that were used in our feature selection pipeline. MIC and Spearman are the two correlation types that can be used to create graphs. Four graph centrality types have been implemented by our pipeline: degree, closeness, betweenness, and eigenvector. Each of these centralities have a different interpretation of what nodes in a graph are important in the network. We will use a sensitivity analysis to investigate the potential effects of using different correlations and centralities to select important features. Because it is desirable to evaluate the how the ordering of selected features differ, rank-biased overlap will be used as one of the evaluation tools.

CHAPTER FOUR

ARCHITECTURE AND IMPLEMENTATION

The goal of this thesis is to design, implement, and evaluate a modular feature selection pipeline that is able to select features from datasets without a priori knowledge about the data. This pipeline is especially desirable for using with large *p*, small *n* datasets, where traditional statistical methods fail.

4.1 Design Considerations

A main consideration when designing this pipeline was the need for customization. Because we lack context for determining important features, the crux of the pipeline is its ability to use different combinations of the parameters to determine a stable list of results. This is shown in our evaluation chapter and is performed using a sensitivity analysis.

The system was engineered in a modular style to allow the sections to be easily interchanged, allowing a sensitivity analysis where the major components and parameters can be changed with little effort in one script. This model also allows for easier future development. For example, one could simply swap out the graph creation and metric steps for a new analysis step while maintaining the integrity of the pipeline itself.

4.2 Architecture

The feature selection pipeline was structured in four distinct steps, as shown in Figure 4.1: conditioning, graph creation, metric processing, and selection.

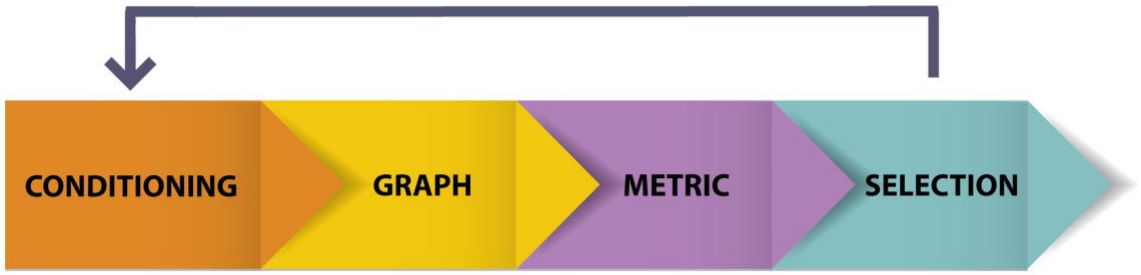

Figure 4.1 Pipeline Architecture – a conceptual diagram of pipeline used for winnowing datasets with network analysis.

4.2.1 Conditioning

The conditioning step transforms the data prior to analysis. The first thing to consider is the existence of false positive OTUs. As discussed previously, sequencing issues can result in false OTUs being reflected in the data. To counteract this problem, the `minimum count` parameter removes all OTUs with a count of `x` or under.

The data is conditioned after OTUs below the minimum count are ruled out. Two types of conditioning have been implemented: the Hellinger transformation and add-one (Laplace) smoothing.

- **The Hellinger transformation** is a distance measurement that is frequently recommended for the ordination of abundance data in ecology [84]–[86]. Hellinger is particularly well suited for ecological data because it excludes the double zero problem, which is where the absence of species at two sites is mistakenly measured as a similarity [85], [87]. It also prevents the case where an increase in total abundance over time shows as a trend [87].
- **Add-one (Laplace) smoothing** is traditionally used in Bayesian classification and natural language processing analyses to prevent probabilities of zero when dealing with unseen events (zero-frequency problems). In this case, it is also used to prevent the double zero problem.

Table 4.1 Parameters relevant to the conditioning step

Parameter	Options	Default Value
Minimum Count	integer value	0
Conditioning Type	hellinger transformation, add-one smoothing	hellinger

4.2.2 Graph Creation

Graphs are then created from the conditioned data. Several steps must be taken before the graphs can be created.

1. A pairwise correlation is performed on the conditioned abundance data, using either the Spearman or MIC as the correlation type. Spearman was chosen as it is a simple similarity measure that has been used in many other ecological studies [3], [21], [24], [57], [69]. MIC was chosen because it is a non-parametric test that is able to identify many different types of associations [27], [57], [88].
2. A check for a user-defined thresholding value between 0 and 1 is performed to evaluate only those correlations that are sufficiently strong.
3. A graph is created, where the nodes are features (OTUs) and the edges are the correlation values between two features. Both weighted and unweighted graphs can be created.

Table 4.2 Parameters relevant to the graph creation step

Parameter	**Options**	**Default**
Correlation Type	spearman, MIC	spearman
Correlation Property	positive, negative, both	both
Weighted	true, false	false
Threshold	decimal value [0,1]	0.2

4.2.3 Metric

The metric step of the pipeline is where the most intensive processing takes place. Network analysis is used to determine a centrality value which implied the importance of features (OTUs). The centrality values of the features are then used to determine a ranked list of features. We implemented the following four graph centrality measures: betweenness, closeness, degree, and eigenvector. These centralities were chosen because they consider different aspects of importance and encode biologically useful relationships [30].

One consideration when dealing with graphs is the question of what should happen if the graph becomes disconnected [75], [79]. To address this issue, the largest connected subgraph is used to calculate the graph centrality. Users may decide that their analysis is only valid if a certain percent of the nodes are connected. For this reason, the pipeline also checks to see what percentage of the whole network is made up by this largest connected subgraph. The corresponding user parameter, `percent connected`, is used to decide if the pipeline should continue.

A second metric check occurs when using the eigenvector centrality. Due to the power iteration method used in computing the eigenvector centrality, it is possible that this centrality will not converge on an answer. Because of this, there is a check that terminates the program if the graph fails to converge. If either the connected check or the eigenvector convergence check fail, the features that have already been selected are returned and the program is terminated.

Table 4.3 Parameters relevant to the metric step

Parameter	Options	Default
Centrality Type	betweenness, closeness, degree, eigenvector	degree
Percent Connected	integer value [0,100]	0

4.2.4 Selection and iterating through the pipeline

The results from the metric stage are passed into the selection step in the form of a ranked list of features (OTUs) and centrality value. The `n` features with highest centrality values are then selected from this list, where `n` is defined by the 'select per iteration' parameter. If fewer than `n` features exist in the list, all features are returned.

The pipeline loops through these steps until the desired number of features have been selected. Once a feature is selected, it is removed from the dataset for the remaining iterations. Because the centrality value is the measure of importance of an OTU, there is no way to choose a single OTU in the case of ties. Therefore, if multiple features with the same centrality value are selected on the last iteration of the pipeline, all values are returned. This concept of iterations allows us to test the potential differences of creating the graph once to select the top `n` most important features, versus creating the graph and selecting between `1` and `n-1` most important

features until `n` features are selected.

For example, if the user wants to select 50 features total and selects 5 per iteration, the pipeline process is run 10 times. For the first iteration of the pipeline, the full dataset is used to condition and build the graph. The top 5 selected features are selected and removed from the dataset the process will be run again. This will continue until the 10^{th} iteration finishes and all 50 features are selected. In some cases, when selecting the 50^{th} feature, there are multiple features with the highest remaining centrality. Because there is no justifiable way to determine the most important of these, all tied features are included in the result set.

The selection process is implemented in this iterative fashion so we can observe if the result sets differ when features are selected all at once, or with some features removed in subsequent iterations. More specifically, if a main hub is removed from the graph and a new graph is built with the remaining features, does the new graph's structure cause the selected nodes being different than if they had been selected from the initial graph? It is more efficient with regards to performance to only create the graph once, so ideally there will be no difference in the features selected.

Table 4.4 Parameters relevant to the selection step

Parameter	**Options**	**Default**
Select per iteration	Integer Value	None
Select total	Integer Value	None

Figure 4.2 visualizes the pipeline process a flowchart that is more descriptive than the basic architecture diagram given at the start of this chapter.

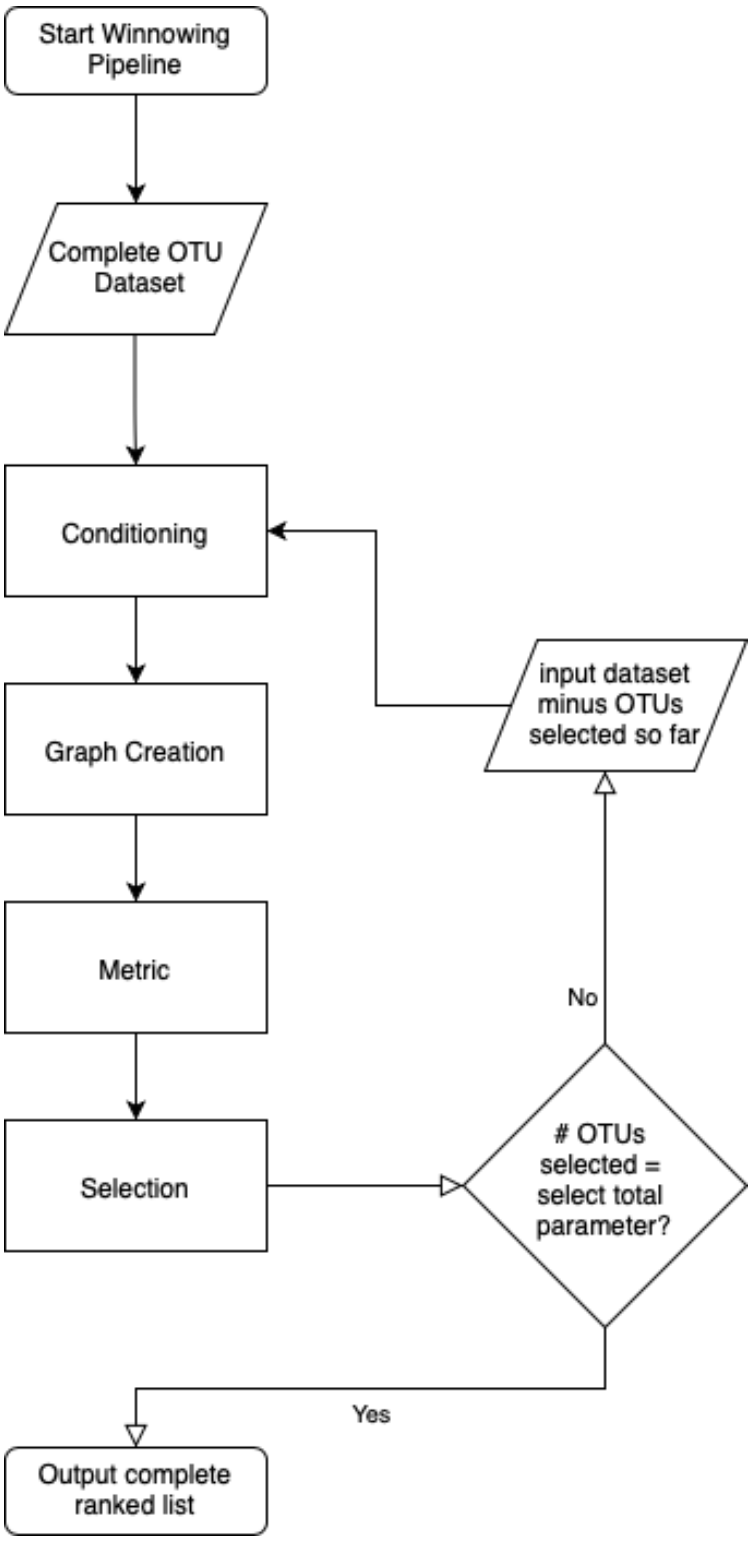

Figure 4.2 – Detailed flowchart depicting of the pipeline to illustrate how the iterations occur in the processing. For each iteration of the pipeline, the input data is the original dataset minus any OTUs that have already been selected by the pipeline.

4.2.5 Output Results

By default, the program outputs three csv files: one containing the list of parameters used; one including the selected feature names, summed abundance number of each feature, and the centrality values of each feature; and an *n*x*m* matrix of the selected features and site abundances. While it is possible to output additional files, such as the graphs that are generated at each iteration, these files were chosen as the default because they contain the most pertinent information to analyze the results without significantly impacting performance.

4.2.6 Complete Pipeline Parameter List

As section 4.2 has demonstrated, the feature selection pipeline is highly parameterizable. Table 4.5 summarizes the parameters available in the pipeline.

Table 4.5 - A list of the parameters available for customization in the pipeline.

Parameter	Description	Options	Default Value
metric	This is graph centrality for our analyses, but another metric, such as PCA, could be implemented and used here.	graph_centrality	graph_centrality
minimum count	The minimum total abundance count of a feature to be considered in the analysis. Any features with this minimum number or fewer will be removed from the dataset before analysis.	positive integer value	0
conditioning	The type of data transformation to perform on the data prior to analysis.	hellinger, add_one	hellinger
select total	The total number of features to select in analysis.	positive integer value, ‘all’	None
select per iteration	the number of features that should be selected at each iteration through the pipeline without replacement.	positive integer value	None
centrality type	When using graph centrality as the metric, this specifies the type of centrality to use to calculate the ‘importance’ of the features.	betweenness, closeness, degree, eigenvector	degree
correlation type	When using the graph metric, this is the type of correlation to use to build the network.	spearman, MIC	spearman
threshold	When using the graph metric, this is the threshold value to remove weak edges. At each iteration, after the network is created using the specified correlation, any edges with correlations less than the threshold will be removed from the network.	positive decimal value $\in [0,1]$	0.2

Table 4.6 continued

weighted	When using the graph metric, the weighting is a parameter that specifies if the network edges should have weights assigned to them. If weighting is used, the centrality will consider the weight value when calculating the centrality values.	boolean	False
percent connected	When using the graph metric, this parameter specifies if the graph should be evaluated if the largest connected subgraph doesn't make up a certain percentage of the entire network. The metric step will only continue if the largest connected subgraph makes up the percentage value specified or higher.	integer [0,100]	0
correlation property	When using the graph metric, this parameter specifies if the network should consider either positive or negative correlations, or both.	positive, negative, both	both

4.3 Pipeline Implementation

4.3.1 Software Libraries

The pipeline is implemented in `python 3`, using several supporting modules. `minepy (v1.2.1)` is used for calculating the MIC correlation, and `pandas` is used for calculating the Spearman correlation as well as doing much of the dataset manipulation.

`networkx (v2.1)` is used in the graph creation and metric steps. This module was is to create the networks from given adjacency matrices, calculate the graph centralities, and find the largest connected component.

`matplotlib (v2.0.2)` is useful for displaying plots of the metric results. The third-party software `cytoscape (v3.5.1)` is valuable for creating stylized graphs from `GRAPHML` files generated by the python pipeline, which are used as visual aids throughout the project. To run the sensitivity analysis, we used a Linux server with 56 cores and 630GB RAM.

The heat maps in Chapter 5 were created with the `clustermap` function in the `seaborn` module (v0.9.0). This function creates a heat map with the option to employ hierarchical clustering to organize the map using the default metric, Euclidean distance, to calculate the

distances for the clusters. Hierarchical clustering was not utilized in these figures because the groupings made it more difficult to compare the different parameter combinations.

4.3.2 Dataset Description

The sensitivity analysis described in Chapter 5 was performed on a previously published taxonomic abundance dataset. The data was derived from 16S rRNA DNA barcoding and subsequent OTU classification of fescue prairie soil undergoing smooth brome (*Bromus inermis*) [29]. For reference and completeness, a summary of the procedure that our colleagues in soil science used to create the data set follows.

Samples were taken from plots established through stratified random design interspersed across the sampling area, with locations determined using a random point generator, ArcMap (Esri, Redlands, CA). The depth of the A horizon was measured using soil colour and texture changes [29]. Two 5 cm diameter soil cores from the top 5 cm from each of the A and B horizons from each plot were extracted using an AMS soil corer (AMS, Inc. American Falls, ID) then further combined and frozen at -20 degrees Celsius. Roots were picked out and organic carbon and total nitrogen were determined for a separate study. An Ultraclean Soil DNA extraction kit (MoBio, Calsbad, CA) was used to extract DNA from 0.5 g of 2 mm sieved soil. Amplification of extracted DNA was performed in triplicate using the 16S rRNA bacterial universal primer set 515F/806R [89]. Ion Torrent sequencing using a 318 v2 chip kit (Life Technologies, Thermo Fisher Scientific, Watham MA, USA) was performed by Contago Strategies (Saskatoon, SK). Sequences were processed for quality control and subsampled to 2550 reads [29] using the `mothur` software package [90].

Taxonomic assignment of OTUs was completed using a naive Bayesian classifier implemented in `mothur` against the Greengenes 2011 database [29]. Bacterial OTUs were separated from fungal and archaeal into an abundance data file for the final 56 samples. The short amplicon length of the 16S bacterial UT rRNA, 291 bp [89] makes Ion Torrent a favourable process of sequencing due to the proven speed of this method [91].

Figure 4.3 illustrates the exponential distribution of the data, after OTUs with a minimum frequency count of three or less had been removed to safeguard against sequencing errors. Note that these histograms are shown with the y-axis scale as logarithmic and the x-axis scale as linear.

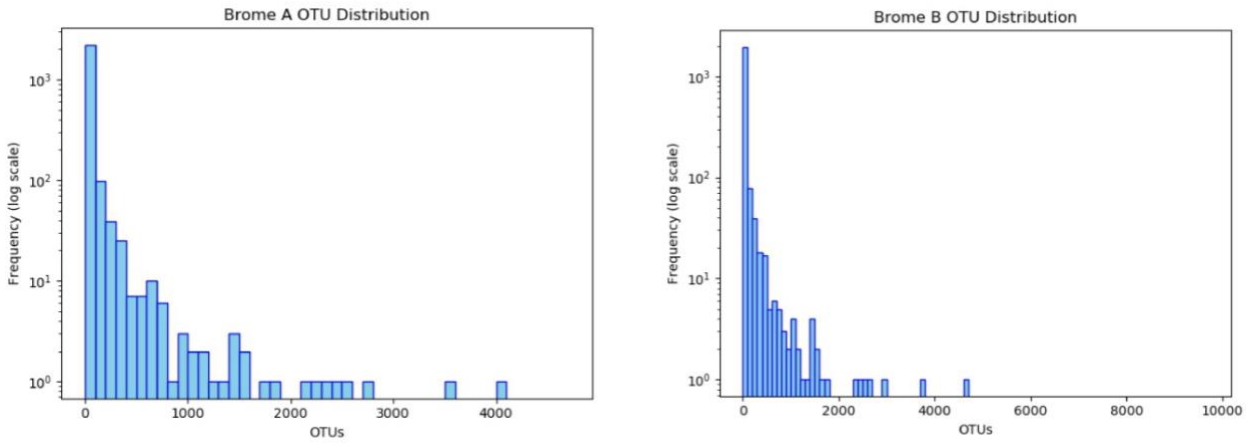

Figure 4.3 Data distribution of Brome A horizon and Brome B horizon. The y-axis is shown as a log scale to better visualize the data. The sensitivity analysis used a combined dataset with both horizons for analysis.

An exponential decay was exposed when plotting the distribution of the data, revealing that there are very few OTUs that are highly abundant. This situation illustrates how simple abundance can be a poor operationalization of importance. Using different definitions of importance, as suggested in our pipeline, may find valuable OTUs that are not overly abundant but still play an important ecological role.

We combined the data from the A and B horizons into one dataset to use as input into our pipeline analysis. This resulted in a dataset of 6747 bacteria, 1339 archaea, and 4014 fungi over 109 samples.

CHAPTER FIVE

EXPERIMENTAL SETUP AND RESULTS

5.1 Sensitivity Analysis

We used the pipeline to perform a sensitivity analysis to determine how different network parameters impacted the ranking of OTU importance. This sensitivity analysis was a key component of the project as it allowed for evaluation of the results returned by the pipeline. We were interested in examining how different parameters used in the pipeline would affect the consistency of the results returned. This provided insight into the reliability of single runs of the pipeline. One potential outcome of this was finding the least computationally intensive combination of parameters that produced a reliable set of results. This may also lead to using multiple runs of the pipeline together to produce the most consistent set of results in downstream analysis.

We focused on varying the following parameters: the centrality type, correlation type, conditioning type, number of features selected per iteration, and thresholding value. Table 5.1 summarizes the values that were used for these parameters.

Table 5.1- A summary table describing the parameters used in the sensitivity analysis.

Parameter	Values
centrality type	betweenness, closeness, degree, eigenvector
correlation type	spearman, MIC
conditioning type	hellinger, add one
selected per iteration	1, 4, 16, 64, 128
threshold	0.2, 0.3, 0.4

Permuting over these parameters resulted in 240 different runs of the pipeline. This has been illustrated using four trees placed side by side in Figure 5.1. Each tree represents the parameters used for one of the four centrality types.

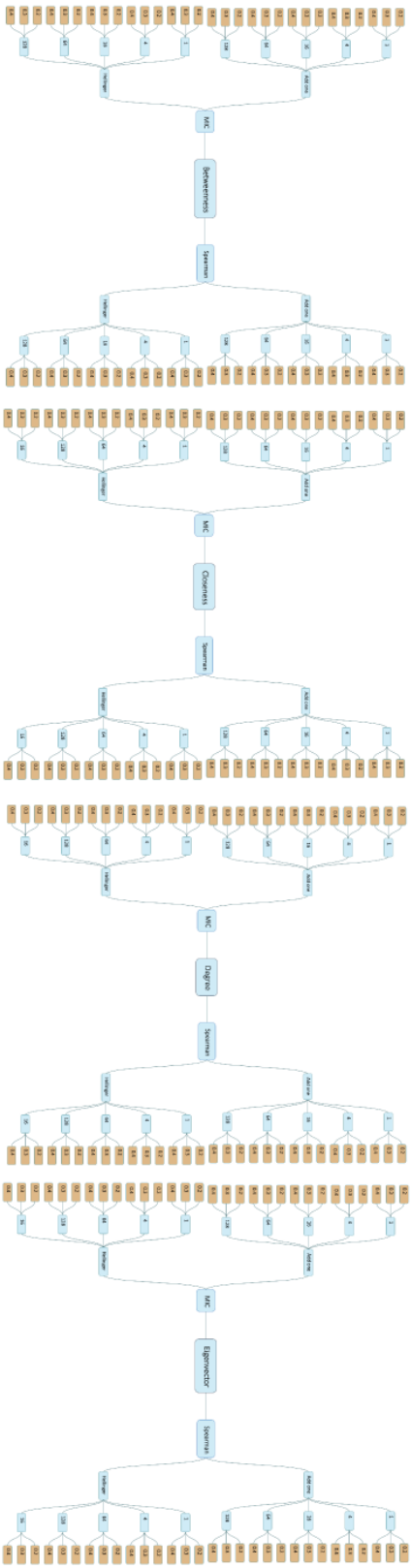

Figure 5.1- Tree representation of all combinations used in the sensitivity analysis.

Figure 5.2 shows a closer look at the tree representation of the combinations of parameters for betweenness centrality. Each leaf of the tree represents one run of the pipeline, with the parameters from the parent branches.

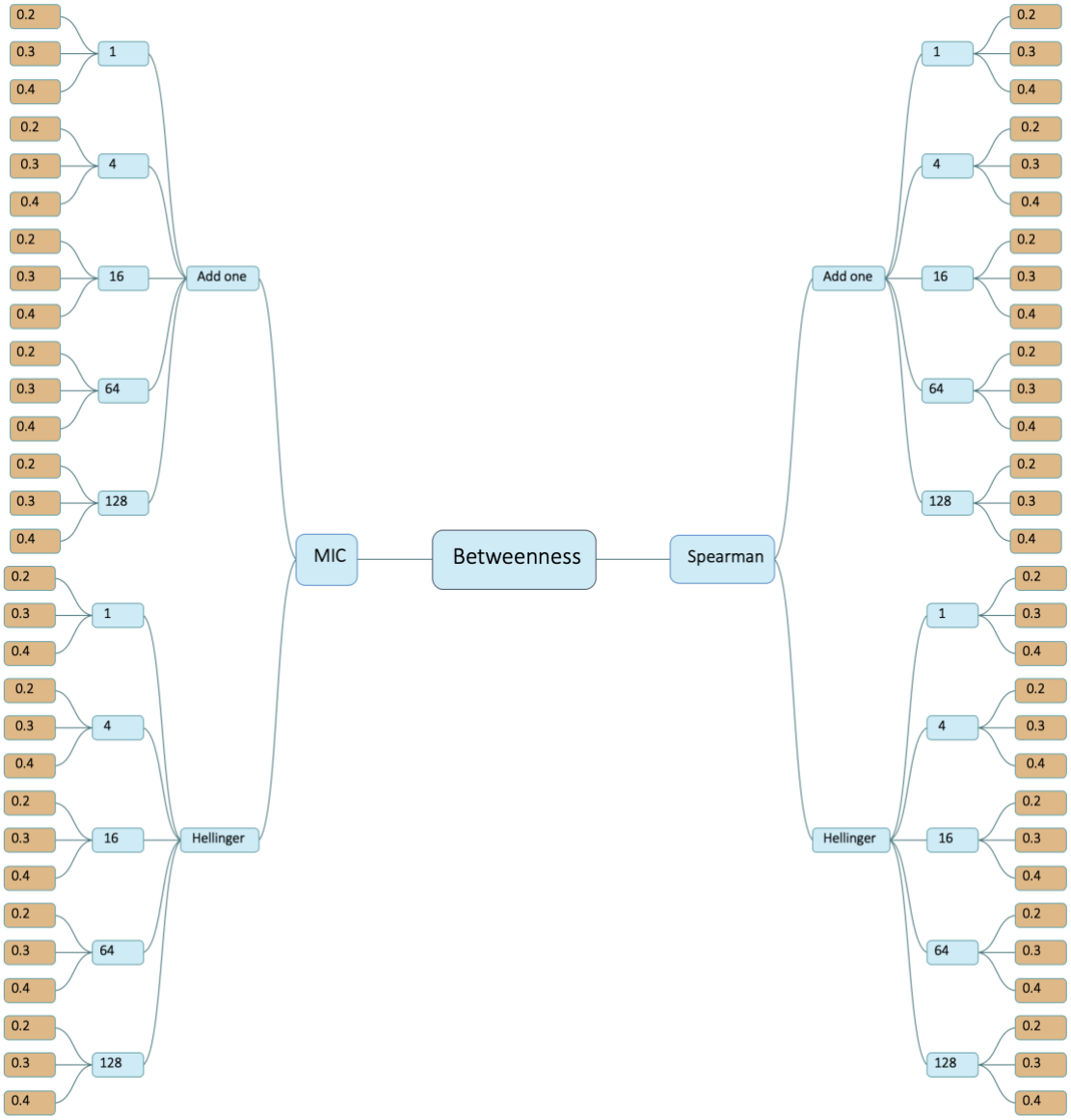

Figure 5.2 - A closer look at the combinations of pipeline parameters for betweenness centrality.

The remainder of this section describes in detail the parameters that were used, why they were used, and explains any other processes that took place during that portion of the pipeline.

5.1.1 Conditioning

Before these conditioning methods were performed, we first removed all OTUs with a count of three or under for our analysis, using the minimum count parameter. Although there isn't a universally agreed upon method for choosing this value, there is a consensus that due to the prevalence of sequencing errors and the possibility of chimeras, OTUs with low abundances are often sequencing artifacts instead of actual rare species [92], [93]. We used a minimum count of three in this sensitivity analysis because it was recommended by our collaborators in soil science. We employed both Hellinger and Add-one conditioning types in the sensitivity analysis to evaluate if the conditioning type influenced the results of the pipeline.

5.1.2 Graph Creation

The sensitivity analysis was run with the Spearman and MIC correlation types. We chose to include both positive and negative correlations when creating the graphs because we wanted to include any relationships that exist, whether they are mutually beneficial or inhibitory relationships. This resulted in an absolute correlation value between 0 and 1.

After the pairwise correlation was calculated for all OTUs, we employed the thresholding parameter to select the strength of correlations included in our analysis. We chose threshold values of 0.2, 0.3, and 0.4 for our sensitivity analysis to observe how the strength of the correlations included in the graphs affected the OTUs that were selected by the pipeline. These values were selected because they provided a variation in graph size, without shrinking the graph to a number of edges that was too small to evaluate. These values were determined by empirical pilot testing. We found that a threshold 0.5 resulted in too few edges and graphs that were very disconnected. A threshold of 0.1 seemed too low of a correlation to hold much meaning and resulted in very dense graphs.

5.1.3 Metric

We tested all four of the implemented graph centralities: betweenness, closeness, degree, and eigenvector. The percent connected parameter was not varied in the sensitivity analysis and the value was left at the default of 0.

5.1.4 Selection and Iterations

The sensitivity analysis selected 128 features, and selected 1, 4, 16, 64, and 128 OTUs per iteration. Selecting a total of 128 OTUs was recommended by our associates in soil science. of two up to 128 to use as the different number of OTUs to select per iteration of the pipeline. Due to the existence of ties in centrality value, many of the results returned more than 128 OTUs, with as many as 163 being selected in one case. Alternatively, other runs of the pipeline returned fewer than 128 OTUs. These cases are discussed later in this chapter.

5.2 Evaluation of Sensitivity Analysis

To evaluate the results from the sensitivity analysis, we reviewed how stable the results returned from the pipeline under different conditions were. We employed two methods to evaluate result consistency: how frequently OTUs were selected across all the pipeline runs, and how similar the result sets were between the pipeline runs, including ordination of results.

A main goal of the sensitivity analysis was to determine if the results differed when selecting a different number of features per iteration. As it is far less computationally intensive to create the graph only once instead of many times, the hope was that the results would be consistent no matter how many features were selected at once.

5.2.1 Result Evaluation

We evaluated the number of times the pipeline returned a full, 128 feature result set. There are a number of reasons that the centralities may not have returned the full result set:

1. The graph may have been disconnected. In this case, the largest connected subgraph was used in the calculation. If the largest connected subgraph had fewer vertices than the `select per iteration` value, then the full number of features could not have been returned.
2. The centrality value of a vertex in the created graph may have equaled zero.
3. The eigenvector centrality's power iteration method may have not converged on an answer.

Figure 5.3 separates the results by centrality type, with a potential 60 runs for each type.

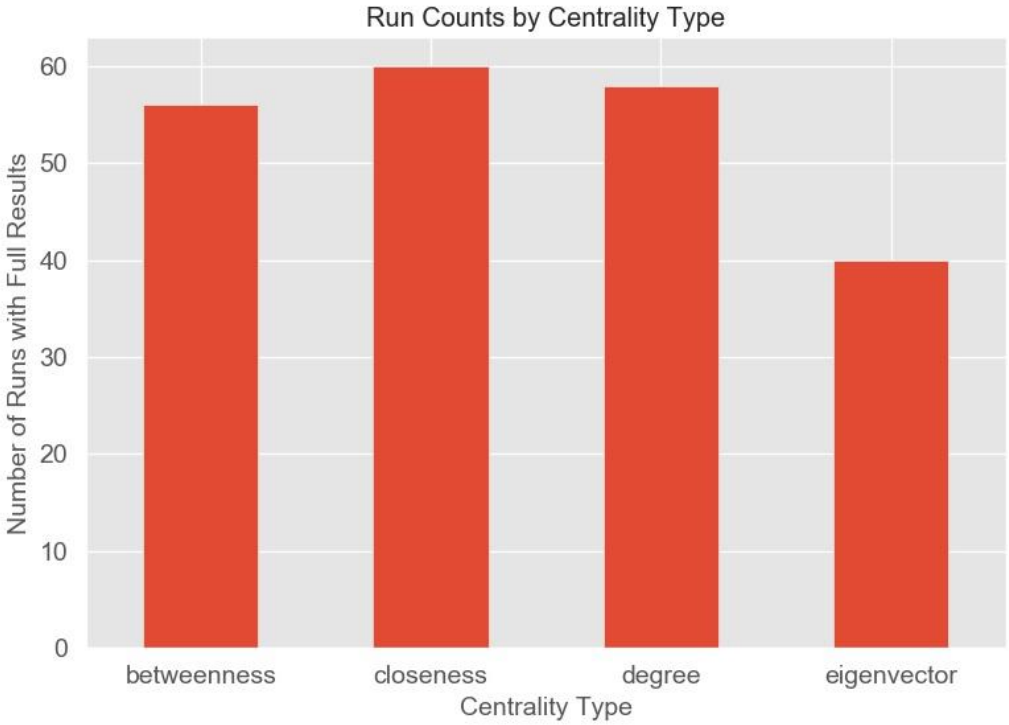

Figure 5.3 - Number of pipeline runs returned with full results, broken down by centrality type

Closeness was the only centrality that returned full results across all 60 combinations of parameters. As mentioned previously, the centrality was calculated on the largest connected subgraph. Because closeness is a calculation of how close a node is to all other nodes in the graph, the centrality value would never be equal to 0. We can infer that due to the way that the OTUs were selected, every iteration of the pipeline was always able to create a graph that was large enough to select the required number of OTUs.

Betweenness failed to return full results four times. The parameters of these runs are described in Table 5.2.

Table 5.2 - Parameters for the pipeline runs for betweenness centrality that did not return complete result sets.

Correlation	Threshold	Select Per Iteration	Conditioning	# Features Selected
MIC	0.4	1	Add one	100
MIC	0.4	4	Add one	110
MIC	0.4	16	Add one	118
MIC	0.4	64	Add one	93

The incomplete results may indicate that there were a few central vertices that connected shortest paths to other vertices. It may also indicate that the graph was too disconnected to return a full set of features.

We examined the last entry in Table 5.2 more closely to determine the cause in this particular case. Pipeline output from this run showed that during the second iteration, the largest connected subgraph made up only 24% of the total graph, with 54 vertices of a total 221 vertices. Figure 5.4 shows the graph that was created from the second iteration when selecting 64 features per iteration.

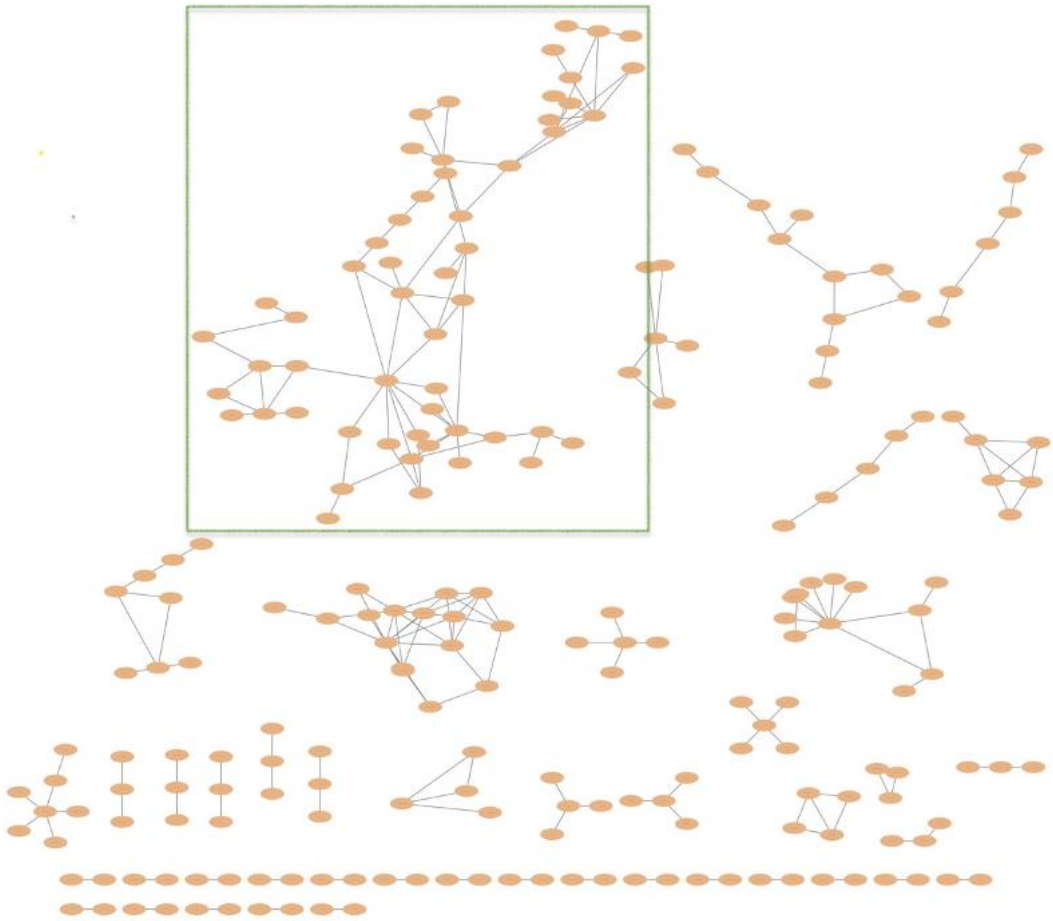

Figure 5.4 - The graph created to calculate betweenness centrality in one iteration of the pipeline.

As shown in Figure 5.4, the graph has many connected subgraphs. Betweenness centrality was calculated on only the largest connected subgraph, which is shown inside the green box in Figure 5.4. Table 5.3 details the betweenness centrality values of the features in this subgraph. Only 29 of these features had a centrality value greater than zero. These 29 features were added to the 64 previously selected, to return the 93 total features selected.

Table 5.3 - A list of betweenness centralities for the features.

Feature Id	Centrality Value	Feature Id	Centrality Value
bac.Otu3880	0.493735	bac.Otu0244	0.007378
bac.Otu5714	0.335269	bac.Otu3510	0.003991
bac.Otu5711	0.300992	bac.Otu5100	0
bac.Otu5981	0.263292	bac.Otu3121	0
bac.Otu3864	0.261248	bac.Otu1936	0
bac.Otu5899	0.248428	bac.Otu6270	0
bac.Otu6448	0.222303	bac.Otu6562	0
bac.Otu5349	0.209361	bac.Otu5737	0
bac.Otu0728	0.20208	bac.Otu6125	0
bac.Otu2777	0.142235	bac.Otu5828	0
bac.Otu4807	0.125544	bac.Otu4918	0
bac.Otu6430	0.108853	bac.Otu5172	0
bac.Otu4285	0.108853	bac.Otu6410	0
bac.Otu3262	0.104451	bac.Otu4383	0
bac.Otu6671	0.101234	bac.Otu0062	0
bac.Otu4431	0.091437	bac.Otu4688	0
bac.Otu6643	0.074746	bac.Otu3245	0
bac.Otu6338	0.074746	bac.Otu5257	0
bac.Otu5851	0.07402	bac.Otu4879	0
bac.Otu5502	0.07402	bac.Otu6018	0
bac.Otu5087	0.042864	bac.Otu4858	0
bac.Otu4451	0.039248	bac.Otu4477	0
bac.Otu4862	0.037736	bac.Otu4885	0
bac.Otu4083	0.037736	bac.Otu5012	0
bac.Otu4960	0.037736	bac.Otu5573	0
bac.Otu3900	0.021771	bac.Otu2472	0
bac.Otu0119	0.007378	bac.Otu5584	0

Degree centrality returned 58 full result sets. The two that were not returned had the parameters as described in Table 5.4.

Table 5.4 - Parameters for the pipeline runs for degree centrality that did not return complete result sets

Correlation	Threshold	Select Per Iteration	Conditioning	# Features Selected
MIC	0.4	16	Add one	119
MIC	0.4	64	Add one	107

Because degree centrality is simply the number of edges a node has, and it is calculated using the largest connected subgraph, the centrality value should never be 0 when selecting features. Therefore, we can determine that these incomplete pipeline results were caused by the largest connected subgraph being smaller than that number of features to select.

Eigenvector returned the fewest number of full results, at only 40. This was a consequence of the eigenvector power iteration method not converging on an answer, as previously discussed in the background information. This happened at different stages of the pipeline, as seen in Table 5.5.

Table 5.5 - Parameters for the pipeline runs for eigenvector centrality that did not return complete result sets

Correlation	Threshold	Select Per Iteration	Conditioning	# Features Selected
Spearman	0.4	1	Add one	0
Spearman	0.4	4	Add one	0
Spearman	0.4	16	Add one	0
Spearman	0.4	64	Add one	0
Spearman	0.4	128	Add one	0
Spearman	0.4	1	Hellinger	0
Spearman	0.4	4	Hellinger	0
Spearman	0.4	16	Hellinger	0
Spearman	0.4	64	Hellinger	0
Spearman	0.4	128	Hellinger	0
MIC	0.3	1	Add one	44
MIC	0.3	4	Add one	40
MIC	0.3	16	Add one	112
MIC	0.4	1	Add one	29
MIC	0.4	4	Add one	40
MIC	0.4	16	Add one	32
MIC	0.4	64	Add one	64
MIC	0.4	1	Add one	2
MIC	0.4	4	Add one	12
MIC	0.4	16	Add one	32

When using MIC correlation, the pipeline always returned some results, though full results were not obtained for both 0.3 and 0.4 thresholding values. The Spearman correlation at a 0.4 threshold was unable to select any features, due to the eigenvector centrality not converging on the first iteration of the pipeline. We have detailed select cases, as described in Table 5.6.

Table 5.6 - Details of graph created under higher threshold values

Correlation	Threshold	Select Per Iteration	Conditioning	Total Nodes	Nodes in LCS	Number of Edges	% Connected
MIC	0.4	128	Add one	404	323	1 722	80%
MIC	0.4	128	Hellinger	858	818	7 184	95%
Spearman	0.4	128	Add one	4553	4496	174 831	99%
Spearman	0.4	128	Hellinger	4553	4499	176 865	98%

The graphs created with the Spearman correlation had significantly more edges than those created with MIC. This data may suggest that the MIC correlation finds fewer, but more meaningful relationships, as opposed to the undiscerning Spearman correlation.

We also tried increasing the number of power iterations run for the eigenvector centrality from 100 (the default in `networkx`) to 1000 iterations for the parameters described in the last row of Table 5.6. This change allowed the centrality to converge at the cost of additional computing time. Although changing the number of power iterations is not currently a parameter in the pipeline, this could easily be added in future work. The runtime increased by over 400%, from 0.09h with 100 power iterations to 0.39h with 1000 power iterations. Depending on computing power, parameters chosen, and size of the data set, this increase may prove too costly for some scenarios.

5.2.2 Performance Evaluation

We analyzed how the correlation type, conditioning type, and centrality type parameters affected the performance of the pipeline in terms of runtime. Runtime was calculated using the `Time` python module and measured in fractional seconds.

When evaluating the runtime of the pipeline, it was critical that we used only the pipeline results that returned a full set of 128 features. Table 5.7 details the eigenvector centrality pipeline runs that selected one feature per iteration. Examining the run described in the first row of this table, we identified that the graph was unable to converge on the third iteration of the pipeline and the program terminated early. As expected, the runtime was significantly lower when compared to

the pipeline runs that completed fully. Including these results would have skewed the representation of how long it takes to select 128 features one at a time.

Table 5.7 - Timing comparison of eigenvector centrality results when selecting one feature per iteration

Correlation	Threshold	Select Per Iteration	Conditioning	# Features Selected	Runtime (h)
MIC	0.4	1	Add one	2	0.67
MIC	0.3	1	Add one	44	8.9
MIC	0.2	1	Add one	128	44.6
MIC	0.4	1	Hellinger	29	13.3
MIC	0.3	1	Hellinger	128	76.5
MIC	0.2	1	Hellinger	128	75.5
Spearman	0.4	1	Add one	0	0.1
Spearman	0.3	1	Add one	128	22.0
Spearman	0.2	1	Add one	128	22.8
Spearman	0.4	1	Hellinger	0	0.1
Spearman	0.3	1	Hellinger	128	23.6
Spearman	0.2	1	Hellinger	128	25.7

As a first look at performance overview, Figure 5.6 provides a boxplot of pipeline runtimes based on centrality type. The y-axis has been converted to hours to readability.

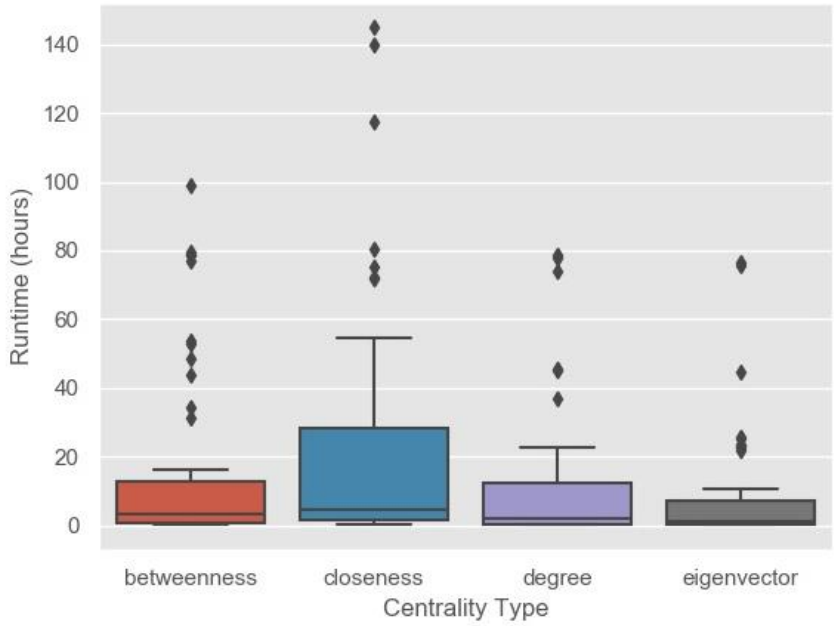

Figure 5.5 - Boxplot of pipeline result runtimes, displayed in hours, broken down by centrality type. This boxplot was created with the Seaborn boxplot method. The box illustrates the quartiles of the data, and the whiskers are determined by 1.5 times the inter-quartile range past the low and high quartiles. Outliers are points outside of the whisker range and are shown as diamond markers.

Figure 5.6 suggests that eigenvector centrality-based analysis had the shortest runtime of the centrality types. However, as previously mentioned, there were a number of eigenvector pipeline runs that did not return full results. For example, there were five eigenvector runs selecting 1 feature per iteration that did not return full results and were eliminated from the timing results. This lack of representation gives a false view of performance for the eigenvector centrality. Though this figure provides a basic summary of timing results, it shows a need to dissect the data further.

Figure 5.7 describes the average runtimes based on each graph centrality measure. The number of features selected per iteration through the pipeline was also considered for each of these parameters to better distinguish the impact on performance. The number of times the pipeline has to build a new graph (i.e. the number of iterations made in the program) had a large impact on performance. That, combined with the situation where runs were excluded because they did not return complete results, made this separation important. Due to the wide span of timing results observed by the features selected per iteration, this figure is shown with a log scale on the y-axis.

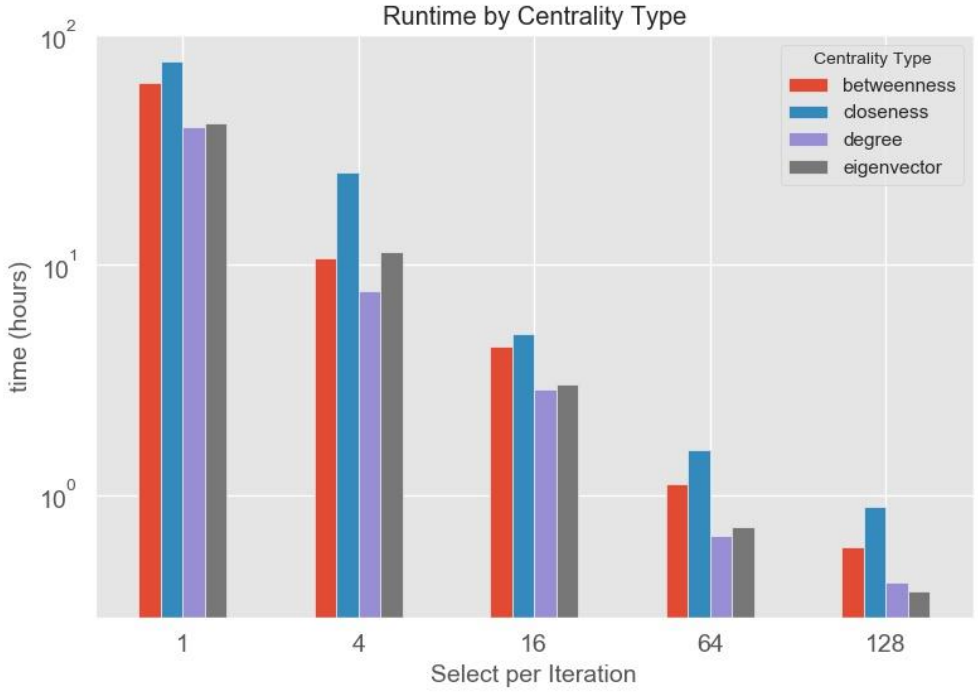

Figure 5.6 - Runtime of full result sets for each 'select per iteration' parameter, broken down by the centrality type.

When breaking the results up by the select per iteration parameter, degree was shown to be consistently the fastest centrality type, with the exception of when selecting by 128 features per iteration. We had expected degree to perform well because of the simplicity of the degree centrality algorithm, in that it only examines the direct relationships of a node.

Eigenvector was a close second with regards to speed and did surpass degree in the 128 select per iteration category. It should be noted that the eigenvector results may not be representative of true performance because one-third of these runs did not return full result sets and were removed from the analysis. However, the runs that selected 128 per iteration returned results in all but two cases for the eigenvector centrality and were faster than all other centrality measures in this group. This provides some assurance that the removal of the incomplete runs may not have skewed the eigenvector performance results shown in the other select per iteration categories.

Closeness was consistently the slowest centrality. This could be attributed to the fact that the closeness algorithm must calculate the distance to all other nodes for each node.

Betweenness was the second slowest centrality tested. Again, this can be attributed to the complexity of the algorithm, with betweenness having to calculate all the shortest paths that go between the given node.

Figure 5.8 (a) compares the runtimes based on correlation type, while Figure 5.8 (b) compares the runtimes based on conditioning type. Due to the wide span of timing results observed by the features selected per iteration, this figure is also shown with a log scale on the y-axis.

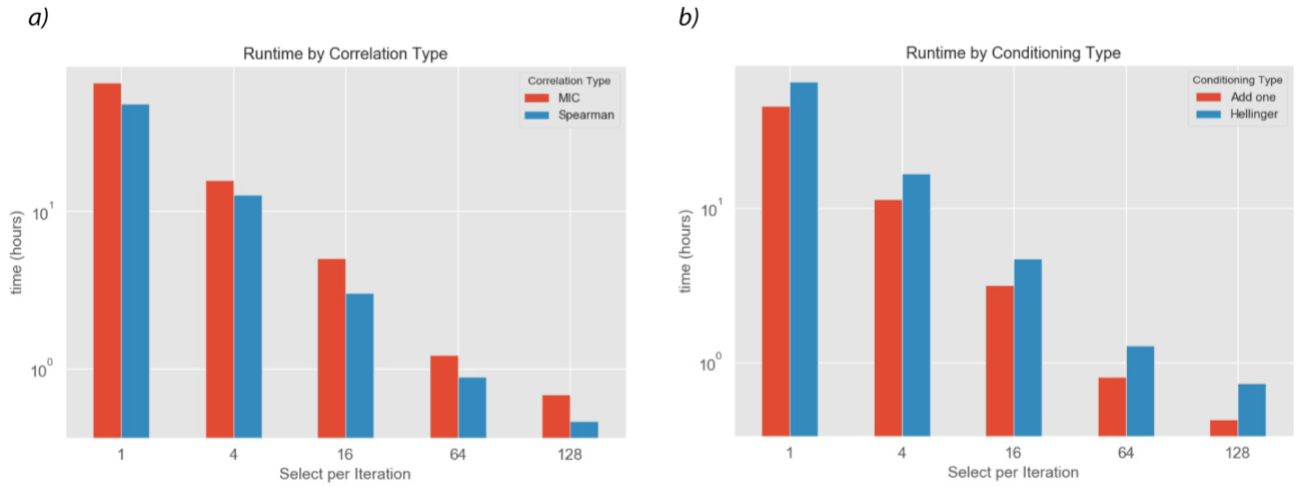

Figure 5.7 - Runtime of full result sets for each select per iteration, broken down by (a) correlation type and (b) conditioning type.

The Spearman correlation slightly outperformed MIC when used as the correlation type. This was expected due to the more complex gridding process that is required by the MIC algorithm. Add one conditioning outperformed the Hellinger conditioning method. This was to be expected because the Hellinger calculation is more complex than the simple additive smoothing performed in the add one conditioning step.

5.2.3 Evaluating Consistency of Top Selected OTUs

The first method for evaluating the results of our sensitivity analysis was to examine the number of times an OTU was selected across all results to determine those taxa which our pipeline considered important regardless of the type of analysis. We then displayed these "most consistent taxa" in a table describing how many times they were selected based on the different pipeline parameters. Note that only pipeline runs that returned at least one OTU were included in this section of the analysis.

To determine the names used in Tables 5.8–5.10, the OTUs were classified by the class and species, where the data was available. When the species was not determined by the sequencing, we used only the class and numbered the entries. Table 5.8 shows the how often each of the 64 most consistently selected OTUs were selected by each centrality/correlation combination. Because there were several eigenvector results that did not return any results and were not included in this analysis, we showed these results as percentages instead of a numerical count.

Table 5.8 - The top 64 OTUs that were consistently selected in the sensitivity analysis.

OTU	Spearman Correlation				MIC Correlation				***Average***
	Degree	Eigenvector	Betweenness	Closeness	Degree	Eigenvector	Betweenness	Closeness	
Actinobacteria 1	100%	100%	100%	100%	100%	100%	100%	100%	100%
0319-6G9 1	100%	100%	100%	100%	100%	90%	100%	100%	99%
Deltaproteobacteria 1	100%	100%	100%	83%	100%	93%	100%	100%	97%
Gammaproteobacteria 1	100%	100%	100%	100%	100%	80%	97%	100%	97%
Actinobacteria 2	100%	100%	100%	83%	100%	87%	100%	97%	96%
Actinobacteria 3	100%	100%	100%	67%	100%	93%	100%	100%	95%
Spartobacteria 1	100%	100%	97%	67%	100%	97%	100%	100%	95%
Actinobacteria 4	100%	100%	100%	67%	100%	87%	100%	100%	94%
Planctomycea 1	100%	100%	100%	67%	100%	87%	100%	100%	94%

Table 5.8 continued

Actinobacteria 5	100%	100%	100%	67%	100%	87%	100%	100%	94%
Actinobacteria 6	100%	95%	100%	67%	100%	87%	100%	100%	94%
fun.s__Gibberella_sp_SB5_2	100%	100%	100%	67%	100%	70%	100%	93%	91%
Solibacteres 1	93%	100%	97%	67%	100%	90%	83%	100%	91%
Actinobacteria 7	100%	100%	100%	97%	83%	83%	83%	83%	91%
Actinobacteria 8	100%	100%	100%	100%	83%	73%	77%	83%	90%
Acidobacteria 1	97%	100%	100%	33%	100%	87%	100%	100%	90%
Acidobacteria 2	100%	80%	100%	47%	100%	83%	100%	100%	89%
Betaproteobacteria 1	83%	90%	87%	67%	100%	83%	100%	100%	89%
fun.unknown_p__Ascomycota_OTU_5	100%	100%	100%	100%	83%	57%	83%	80%	88%
Betaproteobacteria 2	87%	100%	90%	67%	97%	73%	90%	100%	88%
Actinobacteria 9	100%	100%	97%	67%	83%	80%	83%	83%	87%
Thaumarchaeota 1	100%	100%	90%	67%	83%	87%	80%	83%	86%
Sphingobacteria 1	83%	95%	93%	67%	97%	77%	87%	90%	86%
Alphaproteobacteria 1	73%	100%	100%	67%	90%	73%	87%	90%	85%
5B-18	100%	100%	100%	93%	83%	77%	47%	80%	85%
Actinobacteria 10	100%	100%	100%	67%	83%	60%	83%	83%	85%
Chloracidobacteria 1	100%	85%	87%	67%	93%	77%	87%	77%	84%
Actinobacteria 11	100%	95%	97%	67%	83%	73%	80%	80%	84%
Spartobacteria 2	67%	90%	90%	57%	97%	73%	97%	100%	84%
bac.Otu5588	100%	100%	100%	80%	77%	77%	60%	77%	84%
Acidobacteria 3	73%	85%	83%	33%	100%	97%	93%	100%	83%
Actinobacteria 12	100%	100%	100%	67%	83%	60%	80%	77%	83%
fun.s__oat_root_associated_euascomycete_00015	67%	85%	97%	67%	93%	47%	100%	93%	81%
Acidobacteria 4	83%	70%	60%	40%	100%	87%	100%	100%	80%
fun.s__Exophiala_sp_KL_2011f	67%	100%	100%	67%	87%	60%	90%	77%	81%
fun.s__Myrmecridium_schulzeri	100%	90%	100%	67%	80%	47%	77%	73%	79%
Actinobacteria 13	100%	100%	100%	97%	77%	43%	53%	67%	80%
Actinobacteria 14	100%	95%	77%	67%	73%	77%	57%	90%	79%
Spartobacteria 3	50%	70%	97%	33%	97%	73%	100%	100%	78%
SOGA31 1	73%	20%	97%	23%	100%	87%	100%	100%	75%
Planctomycea 2	100%	75%	67%	33%	97%	73%	83%	90%	77%
Bljii12 1	70%	100%	80%	33%	93%	77%	67%	93%	77%
SOGA31 2	100%	100%	100%	100%	47%	60%	40%	63%	76%
Actinobacteria 15	43%	20%	90%	50%	100%	83%	97%	100%	73%
Chloracidobacteria 2	63%	35%	90%	43%	93%	67%	97%	90%	72%
Gemmatimonadetes 1	40%	60%	93%	37%	93%	70%	93%	97%	73%

Table 5.8 continued

bac.Otu6342	50%	50%	80%	33%	93%	80%	97%	97%	73%
Actinobacteria 16	100%	100%	100%	67%	70%	57%	37%	67%	75%
PRR-12 1	100%	100%	90%	37%	83%	53%	57%	77%	75%
Alphaproteobacteria 2	100%	100%	100%	67%	60%	60%	37%	67%	74%
Betaproteobacteria 3	100%	100%	93%	67%	57%	63%	40%	70%	74%
Chloracidobacteria 3	67%	80%	27%	33%	100%	60%	100%	100%	71%
Actinobacteria 17	100%	100%	100%	67%	57%	43%	53%	53%	72%
SOGA31 3	100%	100%	100%	67%	60%	53%	30%	53%	70%
Gemmatimonadetes 2	100%	100%	93%	67%	37%	53%	43%	70%	70%
Actinobacteria 18	100%	95%	63%	33%	83%	70%	37%	80%	70%
Deltaproteobacteria 2	67%	100%	93%	67%	53%	50%	63%	70%	70%
Gemmatimonadetes 3	100%	100%	100%	100%	20%	40%	43%	53%	70%
Betaproteobacteria 4	100%	95%	100%	53%	47%	47%	37%	70%	69%
Deltaproteobacteria 3	33%	60%	90%	37%	80%	73%	87%	77%	67%
Alphaproteobacteria 3	40%	40%	77%	30%	87%	73%	90%	83%	65%
Actinobacteria 19	67%	100%	100%	67%	40%	50%	53%	57%	67%
fun.s__uncultured_Phialophora	50%	65%	73%	63%	83%	53%	77%	57%	65%
Actinobacteria 20	33%	50%	67%	33%	87%	60%	83%	87%	63%
AVERAGE	86%	89%	92%	63%	85%	72%	80%	86%	

This table shows that the betweenness centrality used with the Spearman correlation provided the most consistently selected OTUs. Most of the results are at or above an 80% consistency threshold. Closeness centrality with the Spearman correlation and eigenvector centrality with MIC are the exceptions, performing at the lower consistencies of 63% and 72% respectively.

Table 5.9 limits the results from Table 5.8 to show only the OTUs that were returned by the pipeline in at least 90% of the result sets. The results have again been broken out by the correlation and centrality parameters and show the percentage of results in which they appeared.

Table 5.9 - OTUs that were returned in at least 90% of all pipeline runs under the sensitivity analysis.

OTU	Spearman Correlation				MIC Correlation				***% OTU returned by pipeline***
	Degree	Eigenvector	Betweenness	Closeness	Degree	Eigenvector	Betweenness	Closeness	
Actinobacteria 1	100%	100%	100%	100%	100%	100%	100%	100%	100%
0319-6G9 1	100%	100%	100%	100%	100%	90%	100%	100%	99%
Deltaproteobacteria 1	100%	100%	100%	83%	100%	93%	100%	100%	97%
Gammaproteobacteria 1	100%	100%	100%	100%	100%	80%	97%	100%	97%
Actinobacteria 2	100%	100%	100%	83%	100%	87%	100%	97%	96%
Actinobacteria 3	100%	100%	100%	67%	100%	93%	100%	100%	95%
Spartobacteria 1	100%	100%	97%	67%	100%	97%	100%	100%	95%
Actinobacteria 4	100%	100%	100%	67%	100%	87%	100%	100%	94%
Planctomycea 1	100%	100%	100%	67%	100%	87%	100%	100%	94%
Actinobacteria 5	100%	100%	100%	67%	100%	87%	100%	100%	94%
Actinobacteria 6	100%	95%	100%	67%	100%	87%	100%	100%	94%
fun.s__Gibberella_sp_SB5_2	100%	100%	100%	67%	100%	70%	100%	93%	91%
Solibacteres 1	93%	100%	97%	67%	100%	90%	83%	100%	91%
Actinobacteria 7	100%	100%	100%	97%	83%	83%	83%	83%	91%
Actinobacteria 8	100%	100%	100%	100%	83%	73%	77%	83%	90%
Acidobacteria 1	97%	100%	100%	33%	100%	87%	100%	100%	90%
AVERAGE	99%	100%	100%	77%	98%	87%	96%	97%	

Categorizing these top results by correlation further confirms that the set of OTUs returned by the closeness/Spearman and eigenvector/MIC combinations were both less consistent than the others, as seen in the broader results of Table 5.8.

Table 5.10 identifies the taxonomy of the top 16 results; the same OTUs from Table 5.9, which are selected in 90% or more of the pipeline runs. All of the selected OTUs are in the Bacteria kingdom and the majority are in the Actinobacteria phylum and Actinobacteria class.

Table 5.10 - The taxonomy of the nine most consistently returned OTUs.

OTU	Taxonomy						
	Kingdom	Phylum	Class	Order	Family	Genus	Species
bac.Otu5270	Bacteria	Actinobacteria	Actinobacteria	MC47			
bac.Otu6078	Bacteria	SPAM	0319-6G9				
bac.Otu6245	Bacteria	Proteobacteria	Deltaproteobacteria	Syntrophobacterales	Syntrophobacteraceae		
bac.Otu5330	Bacteria	Proteobacteria	Gammaproteobacteria	Xanthomonadales	Xanthomonadaceae	Dokdonella	
bac.Otu5981	Bacteria	Actinobacteria	Actinobacteria				
bac.Otu4727	Bacteria	Actinobacteria	Actinobacteria				
bac.Otu6544	Bacteria	Verrucomicrobia	Spartobacteria	Spartobacteriales	Spartobacteriaceae	MC18	
bac.Otu6641	Bacteria	Actinobacteria	Actinobacteria	Actinomycetales	Frankiaceae		
bac.Otu6747	Bacteria	Planctomycetes	Planctomycea	Pirellulales			
bac.Otu6413	Bacteria	Actinobacteria	Actinobacteria	Actinomycetales	Pseudonocardiaceae	Pseudonocardia	
bac.Otu6569	Bacteria	Actinobacteria	Actinobacteria	Solirubrobacterale	Solirubrobacterale	Solirubrobacte	
fun.s__Gibberella_sp_SB5_2	Fungi					Gibberella	SB5_2
bac.Otu6448	Bacteria	Acidobacteria	Solibacteres	Solibacterales	Solibacteraceae	CandidatusSolibacter	
bac.Otu5222	Bacteria	Actinobacteria	Actinobacteria	Euzebiales	Euzebiaceae	Euzebia	
bac.Otu5711	Bacteria	Actinobacteria	Actinobacteria	Acidimicrobiales	EB1017		
bac.Otu6684	Bacteria	Acidobacteria	Acidobacteria	Acidobacteriales			

This evaluation method provided a straightforward assessment of how consistent the results are across the correlation and centrality types, and which taxa were most stable. However, it did not take into account where the OTUs were ranked in each list.

5.2.4 Evaluating Consistency Across Parameter Sets

One of the main questions we set out to answer was how using different parameters will change which features are selected and what order they are selected in. Although our first evaluation method showed how consistently individual OTUs were selected across parameter changes, it did not evaluate how consistent the lists were with respect to the ranking of the results. To do so, we needed to examine every pair of the pipeline results to evaluate how the results differed in ranking, as shown in Figure 5.8.

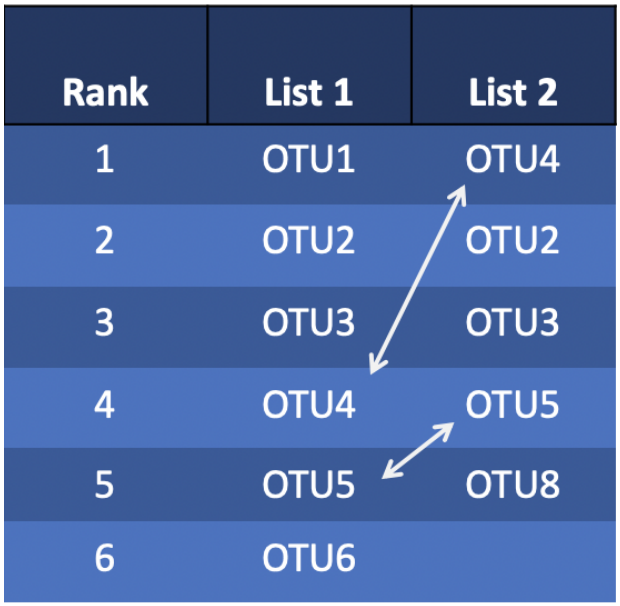

Figure 5.8 An example of the need for Rank Biased Overlap. List 1 and List 2 are disjoint, meaning they do not share all of the same items. Additionally, some of the shared items have different rankings.

As illustrated in Figure 5.9, the simple check that an OTU exists in a list that was performed in the previous section was an incomplete portrayal of the results. Because the features returned by the pipeline are ranked based on centrality value, the ordering of selected features is an important aspect to consider when evaluating the pipeline results.

The Rank Biased Overlap method was chosen to determine the similarity between the output of each of the pipeline runs, taking into account the ranking of results. Rank Biased Overlap was chosen because it handles cases that traditional ordination comparisons do not. Rank correlation methods such as Kendall Tau measure if items in two lists are in the same order. However, these measures do not take into account disjoint lists, where an item appears in one list but not the other. They also do not consider the positional ranking of items in the list. Because these are two critical aspects of analyzing the results of the pipeline results, common methods like Kendall Tau and Spearman rank were not appropriate to use. The implementation of Rank Biased Overlap that was used will address these properties, though the default implementation does not address ties.

Rank Biased Overlap was run pairwise on all results obtained from the pipeline, including those that did not return full results. The results from the Rank Biased Overlap were then used to construct the heat maps in this chapter. The heat map should be interpreted as

follows: results with a strong overlap score (dark blue) imply that those runs have a high commonality with regards to the selected OTUs and the order in which they are selected. Results with little overlap (white) imply that there are minimal common OTUs or ordering between the runs. The consistency of all selected OTUs is illustrated using the heat map shown in Figure 5.10. This was broken out into two heat maps to improve readability: one based on Spearman correlation and one based on MIC correlation.

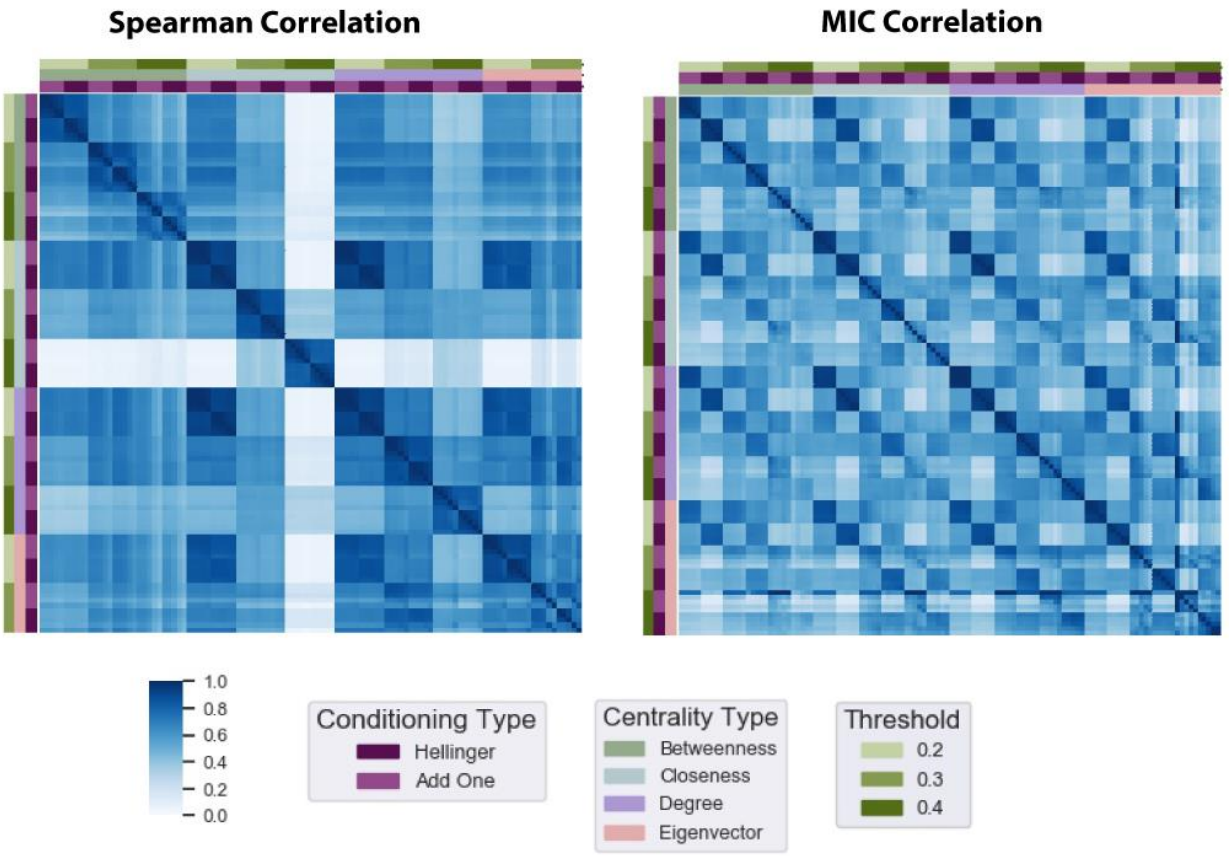

Figure 5.9 Heat map showing the consistency of features selected across the different parameters.

From the heat map in Figure 5.10, it appears that most of the iterations in the sensitivity analysis returned a similar set of OTUs, regardless of the parameter set used in the analysis. The results appeared to be more consistent with the Spearman correlation, with a few exceptions. Much of the dataset exhibited a strong overlap when using the Spearman correlation, with a few distinct

sections with almost no overlap. Upon closer inspection, these were predominantly made up of closeness iterations using the Spearman correlation. It is difficult to determine concretely how these two heatmaps compare only by looking at them. Table 5.11 uses the mean RBO score for each to assess the overall similarity across both the Spearman correlation and MIC correlation.

Table 5.11 - Mean RBO results from comparing Spearman pipeline runs and MIC pipeline runs.

Correlation Type	Mean RBO
Spearman	0.5629
MIC	0.5520

We can presume that the Spearman score was lowered due to the large section shown in Figure 5.10, where closeness had almost no similarity with the others. When closeness was removed from the calculation, the mean RBO score for Spearman increased to 0.6430. This situation will be explored in more detail later in this section.

Figure 5.11 examines the results of the sensitivity analysis when using the Spearman correlation. Each panel considers one type of network centrality measure. The results in each panel are further categorized by how many features were selected per iteration of the pipeline, and the threshold value used when creating the graph.

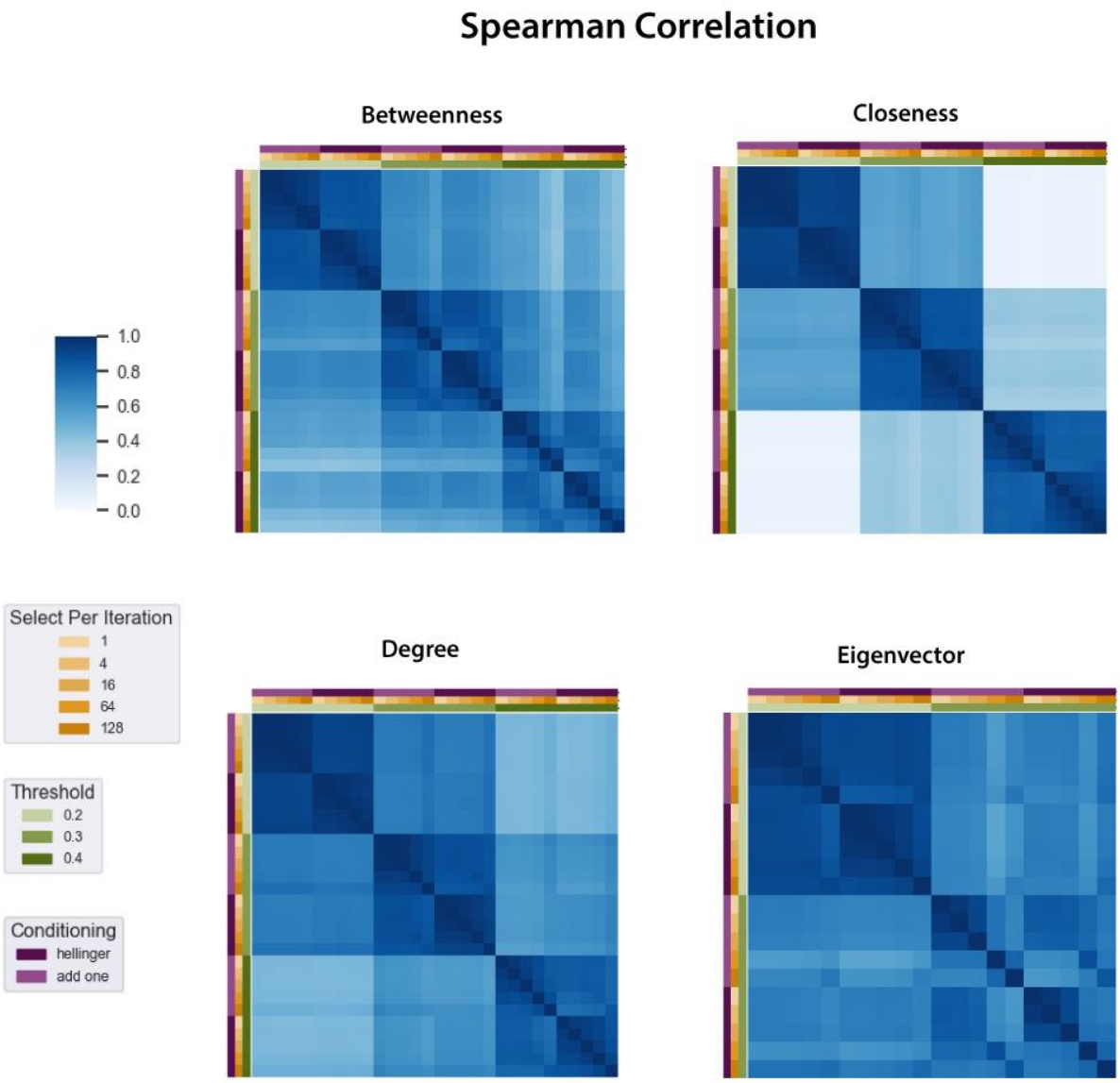

Figure 5.10 Heat map displaying Rank Biased Overlap results for the four different centralities tested using Spearman correlation.

Figure 5.11 illustrates how the thresholding parameter influenced the consistency of results, especially when the closeness centrality was used with the Spearman correlation. The highest threshold of 0.4 had essentially no overlap with the lowest threshold of 0.2 when using closeness as the centrality. It is important to highlight the eigenvector results, as there were fewer results with this graph centrality at the higher thresholds. As previously noted in this chapter, the

eigenvector centrality didn't return any results at a 0.4 threshold when using the Spearman correlation. Figure 5.12 provides a view of the distribution of the Spearman RBO results.

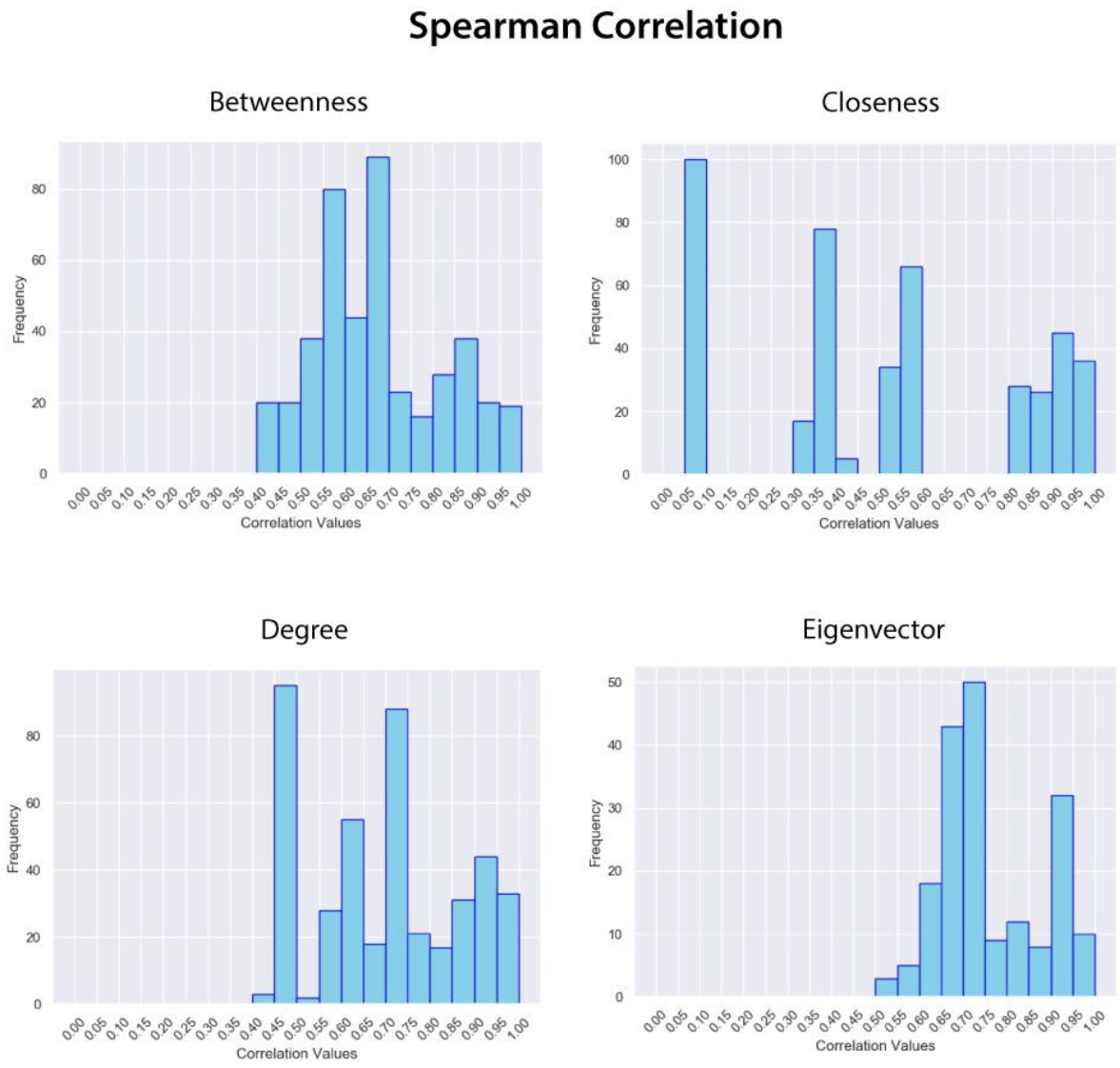

Figure 5.11 – Histogram displaying the distribution of Rank Biased Overlap results for the four different centralities tested using Spearman correlation.

Again, the disparity in closeness results is clear here, with three major groupings distinguishable in the histogram. Eigenvector appears the most similar, though as observed in the heat map, there were no results for the 0.4 threshold pipeline runs. Though this graph may be

simpler to see the overall strengths of correlation, it is missing some important information that was shown by the heat maps.

Figure 5.13 shows the results of the sensitivity analysis when using the MIC correlation. Similar to Figure 5.11, the panels in this figure are based on graph centrality and categorized by number of features selected per iteration of the pipeline, and threshold value used when creating the graphs.

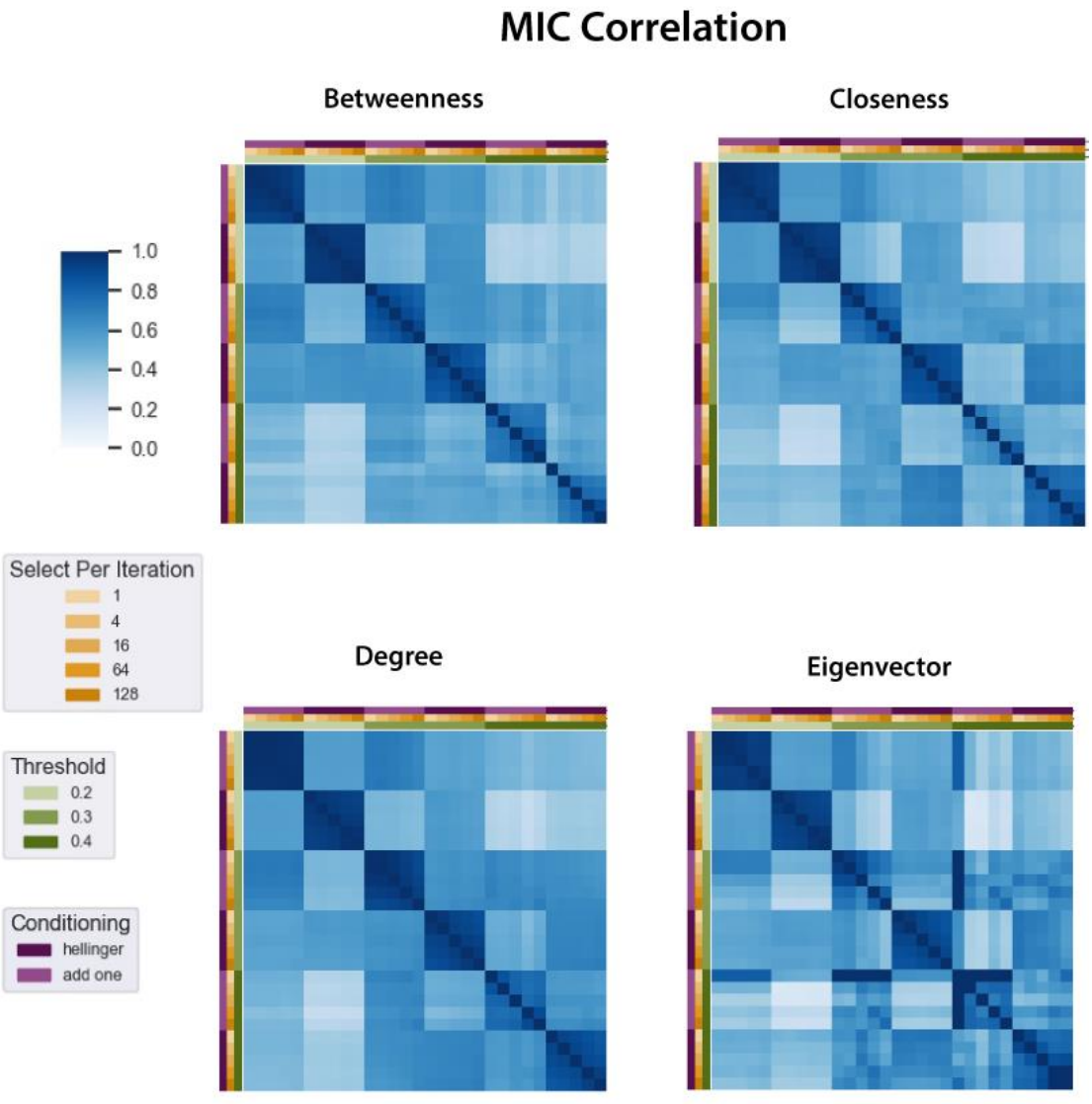

Figure 5.12 Heat map displaying Rank Biased Overlap results for the four different centralities tested using MIC as the correlation metric.

The eigenvector centrality was able to converge on an answer when using the MIC correlation. This is notable when compared to the Spearman correlation, which was not able to converge at the high threshold values, as shown in Figure 5.10. The evident checkered pattern seen in the MIC results in Figure 5.12 is due to the data conditioning parameter. Add-one smoothing and the Hellinger transformation were the two types of conditioning used in the sensitivity analysis. Though the checkered pattern is sometimes observable in the Spearman heat maps (Figure 5.10), it is much more distinct in the MIC results. This indicates that the OTUs selected are more sensitive to change based on the conditioning parameter changes when using MIC to create the graphs than when Spearman was used. Figure 5.13 shows a histogram of the RBO results for the MIC correlation.

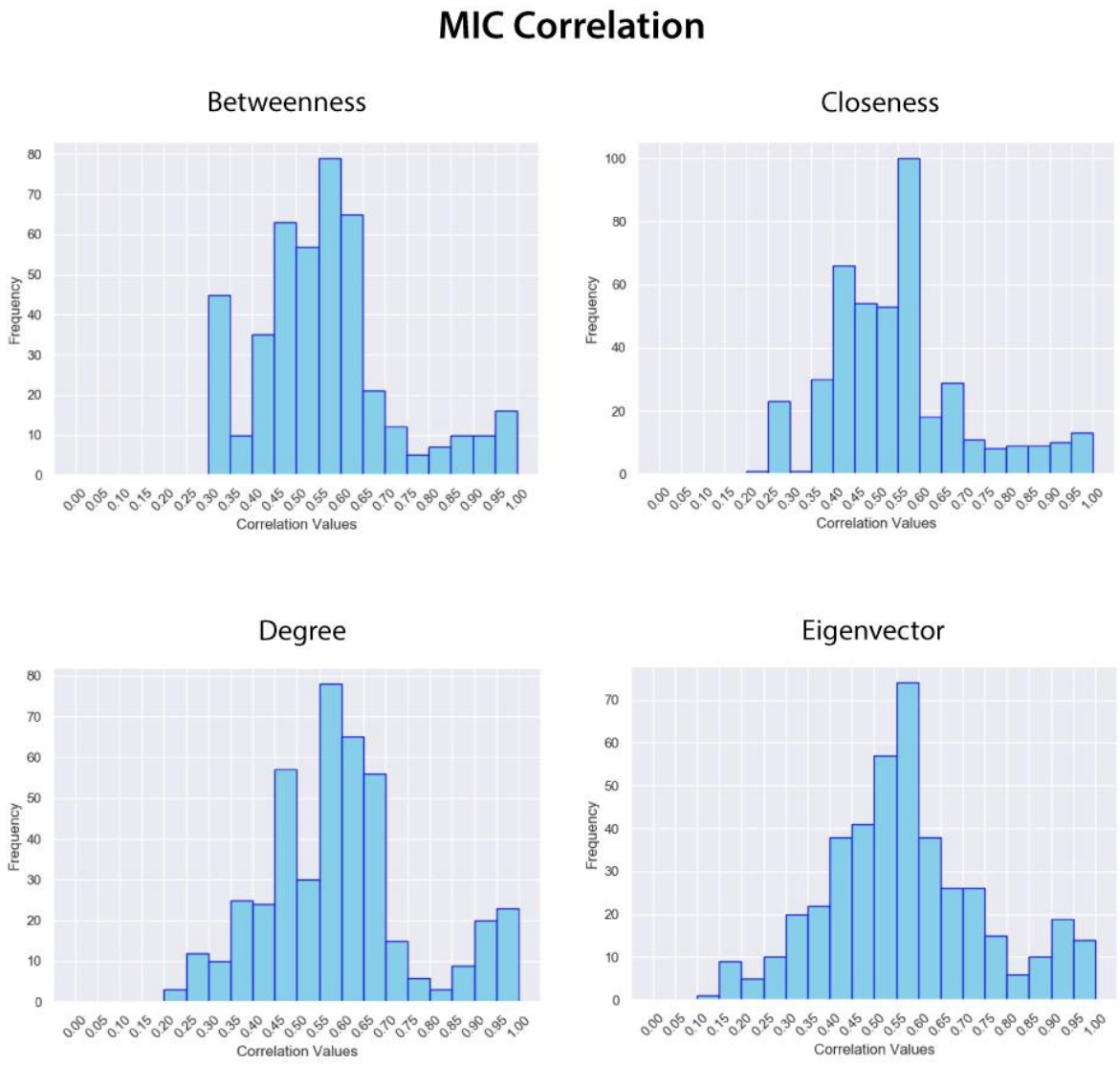

Figure 5.13 – Histogram displaying the distribution of Rank Biased Overlap results for the four different centralities tested using MIC as the correlation metric.

The MIC results are perhaps easier to interpret in this histogram than in the heat map version. Because so many of the results in the heatmap are a similar shade, it can be difficult to get a proper impression of the values. Here, we can observe that with closeness, for example, the majority of the RBO similarity values are 0.55 and below. This plot is also a more satisfactory representation of the eigenvector RBO results than the Spearman histogram (Figure 5.11) because the MIC results were not missing a large component of results like the Spearman results.

It is apparent from the histogram figures (5.12 and 5.14) that, excluding closeness, the RBO similarity scores from MIC had a wider variation than those of Spearman. While the Spearman RBO scores were never below 0.40, MIC RBO scores were as low as 0.10. Table 5.12 summarizes the mean RBO values for each correlation and centrality type combination.

Table 5.12 - The average RBO score for each centrality type and correlation type.

Correlation Type	Centrality	Mean RBO
Spearman	Betweenness	0.6756
	Closeness	0.5115
	Degree	0.6957
	Eigenvector	0.7573
MIC	Betweenness	0.5641
	Closeness	0.5508
	Degree	0.5968
	Eigenvector	0.5686

With respect to the RBO evaluations, the Spearman correlation outperformed MIC for every centrality except for closeness. Eigenvector has the highest mean score at almost 76% similarity. However, as discussed earlier in this section, the Eigenvector analysis did not return any results under the Spearman correlation for the 0.4 threshold. Therefore, this score is not a good comparative indicator to the others. Degree is a close second with close to 70% similarity, and Betweenness is close behind with almost 68% similarity.

5.3 Summary

The sensitivity analysis varied five parameters and resulted in 240 pipeline runs. The parameters of graph centrality, conditioning type, correlation type, select per iteration, and threshold were varied for each run of the pipeline. Evaluation of the sensitivity analysis was done in four main parts:

1. We first looked at the number of times the pipeline returned a full feature result set. For each centrality type, we investigated those cases that returned fewer than the requested 128 features and provided explanations for why fewer results would have been returned. Closeness was the only centrality to return full result sets for every pipeline run, though degree was a close second, with only two runs without complete results.

2. We briefly evaluated the runtime performance of the pipeline based on different conditioning type and correlation type. We focused our performance analysis by breaking the results down by graph centrality and features selected per iteration.
3. We evaluated the consistency of the top 64 OTUs returned in sensitivity analysis. Betweenness centrality with the Spearman correlation had the highest consistency with an average of 92% of the results selected. Degree centrality with the Spearman correlation was the second highest with an average of 86%.
4. Finally, we examined the similarity of the result sets with respect to ranking using Rank Biased Overlap. Eigenvector centrality with the Spearman correlation had the highest average Rank Biased Overlap score of almost 76%, but we identified issues with this centrality. Degree centrality with the Spearman correlation provided the second highest Rank Biased Overlap score of almost 70%.

CHAPTER SIX

DISCUSSION AND CONCLUSION

6.1 Application of the Pipeline

Combining these different types of evaluation has given us insight into possible advantages and drawbacks of the pipeline parameters. Users may choose to use different parameter values based on their downstream analysis requirements, using the results in Chapter 5 to guide them in selecting parameters. These decisions may be based on considerations such as size of data set, available computing power, overall utility, and if they need to ensure full results returned.

If reliability of pipeline (i.e. returning the full number of results requested) is the top consideration of the analysis, the closeness centrality should be considered. It was the only centrality to return the full set of results every time it was used. With the current implementation of the pipeline, it is not advisable to use the eigenvector centrality if reliability is important. Eigenvector failed to return the full set of requested results one-third of the time. Though eigenvector performed reasonably well with regards to speed when full results were returned, this centrality is by far the most likely to produce incomplete results. As discussed below in section 6.3, future versions of the pipeline could improve the reliability of this centrality and make it a viable option.

When considering the speed of the pipeline, users may base their parameter selections on the computing power they have available. If computing power is a concern, the user may want to select the parameters that performed the fastest in the sensitivity analysis. The degree centrality was shown to be the overall fastest centrality type by the sensitivity analysis. Though the eigenvector centrality also appeared to be a close second with regards to speed, this may have been impacted by the already discussed incomplete results. Though closeness was determined to be the most reliable centrality for returning full results, this reliability is compromised by much slower performance. We also examined the conditioning type and correlation type with regards to speed: add-one smoothing outperformed Hellinger, and the Spearman correlation surpassed MIC.

Users may be inclined to select a centrality based on their biological interpretation described in the background chapter. However, if a user is not concerned about this interpretation, they may want to consider the consistency with which OTUs are selected by the different parameters. When we investigated the 64 OTUs that were selected the most times by the pipeline, we found that betweenness centrality coupled with the Spearman correlation provided the most consistent set of results. This parameter combination selected an average of 92% of these OTUs. Closeness centrality with the Spearman correlation had the lowest consistency, at only 63%.

For an additional view of consistency, we examined how often the results of these centrality/correlation combinations ranked selected OTUs in the same order. This analysis showed that iterations that use the Spearman correlation are generally more consistent than those using the MIC correlation. The exception to this was closeness centrality, whose results when run using Spearman at the highest threshold of 0.4 were very dissimilar to both the other Spearman closeness iterations, as well as the other centralities with Spearman. This supports the low consistency score found by the other measure of consistency. With regards to ranking, degree and betweenness coupled with the Spearman correlation were identified as the most consistent centralities, with 70% and 68% ranking consistency respectively. When considering both types of consistency that were evaluated, betweenness and degree with the Spearman correlation were shown to be good options for users to select.

6.2 Contribution

This winnowing pipeline has been used as a key part of a system to predict network evolution. Mamet *et al*. 2019 determined that our pipeline could be used to help find microbes responding to externalities regardless of abundance. A web interface version of our pipeline was run by the researchers as part of their analysis. Figure 6.1 illustrates the full process from Mamet et al.

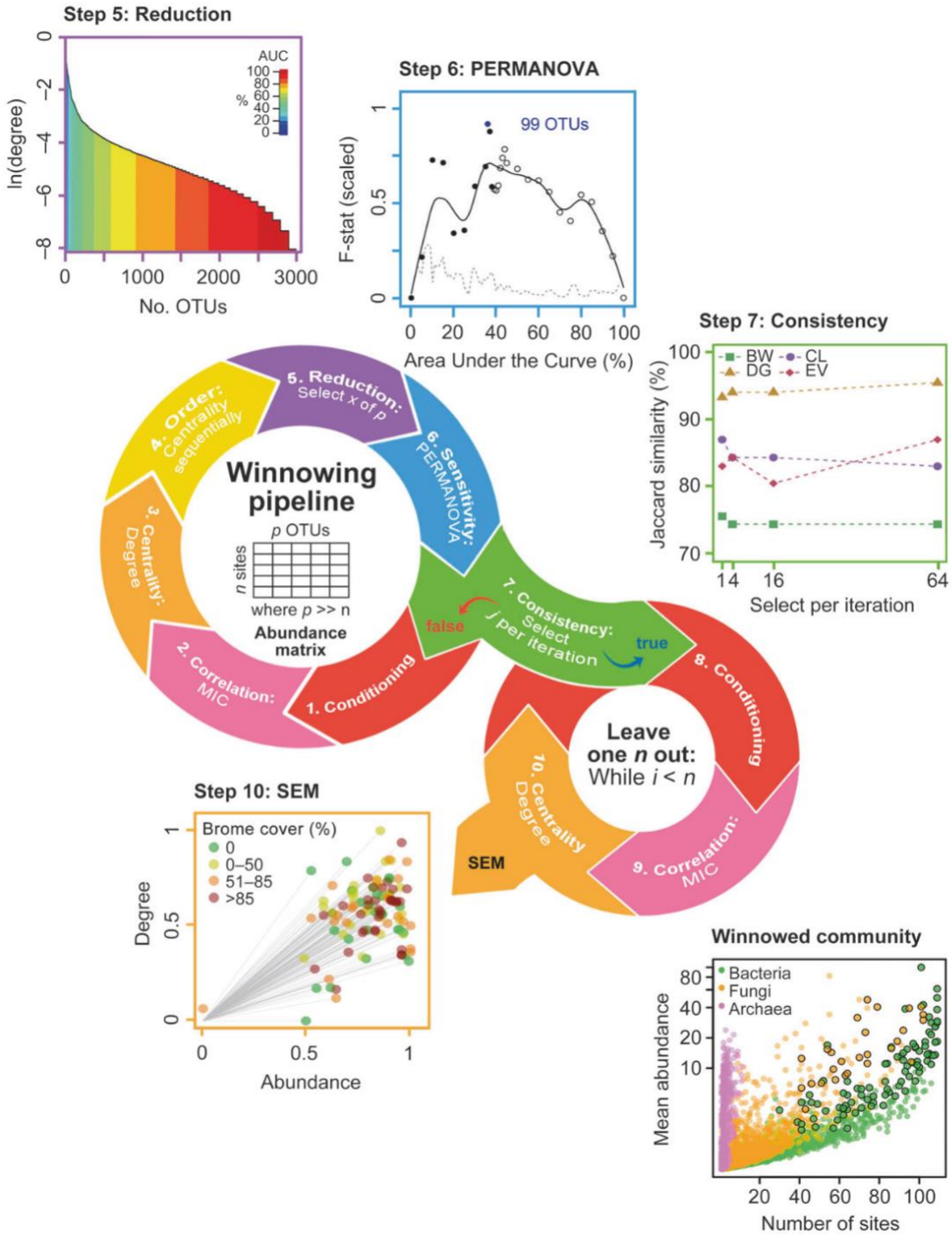

Figure 6.1 A graphical example of the method used by Mamet et al., with our winnowing pipeline serving as several steps of the pipeline, namely steps 1-4, and 7-9.

Our winnowing pipeline was used in three separate stages in the overall process.

1. It was used to narrow down the OTUs used in downstream analysis. The pipeline was run once for each centrality type, using the parameters found in Table 6.1. This resulted in four

sets of pipeline results. For each of the result sets, they reduced the number of selected OTUs based on an area under the curve sensitivity analysis. This reduced list was used in a permutational analysis of variance (PERMANOVA), and OTUs were selected from this to maximize the F-statistic and minimize the standard deviation. After assembling this list of selected OTUs for each of the result sets, they performed a union on the lists that resulted in a set of 115 unique selected OTUs to use in their downstream analysis.

Table 6.1 - Parameters used in pipeline analysis of the first section of the process as described in Mamet et al.

Parameter	Values
threshold	0.2
correlation type	MIC
conditioning type	Add-one smoothing
centrality type	degree, betweenness, closeness, eigenvector
select per iteration	all
select total	all

2. A sensitivity analysis, similar to the one used in this thesis, was performed to assess how sensitive the method was to the number of OTUs selected per iteration, the centrality metric, and the correlation type. The parameters used in this sensitivity analysis are described in Table 6.2. They used the Jaccard similarity index to compare the similarity and diversity of the returned result sets. They found that the pipeline results were relatively insensitive to the number of OTUs per iteration, but it differed among centrality metrics.

Table 6.2 - Pipeline parameters used in sensitivity analysis by Mamet et al.

Parameter	Values
threshold	0.2
correlation type	MIC, Spearman
conditioning type	Add-one smoothing
centrality type	degree, betweenness, closeness, eigenvector
select per iteration	1, 4, 16, 64
select total	128

3. The pipeline was used in a leave-one-out (LOO) analysis to create a dataset that described the importance per sample instead of per OTU, as required by the downstream analysis. For the LOO analysis, a sample by OTU abundance dataset was created from the list of OTUs determined by the first use of our pipeline. The pipeline was run *n-1* times, where

n is the number of samples in the dataset. Each sample was sequentially left out and our pipeline was run without that sample. Each OTU contribution to sample centrality and abundance was summed to quantify the sample centrality and abundance.

The results of the LOO analysis were used in a hypothesis-driven analysis of network evolution among treatments, called structural equation modeling (SEM), that was able to characterize keystone OTUs, illustrated in Figure 6.2. Though the researchers chose to use SEM to link external factors to network evolution, it was noted that our pipeline could also be used as the precursor for other statistical methods, such as generalized linear mixed modeling or additive modeling.

This study made use of the same smooth brome invasion dataset that we used in our sensitivity analysis. A priori knowledge about smooth brome invasions was used; namely, that an invasion lowers plant diversity and increases nitrogen, thereby suppressing dominant bacteria species and increasing abundance of rare species. This marked the first time structural equation modeling was used to predict network evolution [30] and would not have been possible without the help of our pipeline.

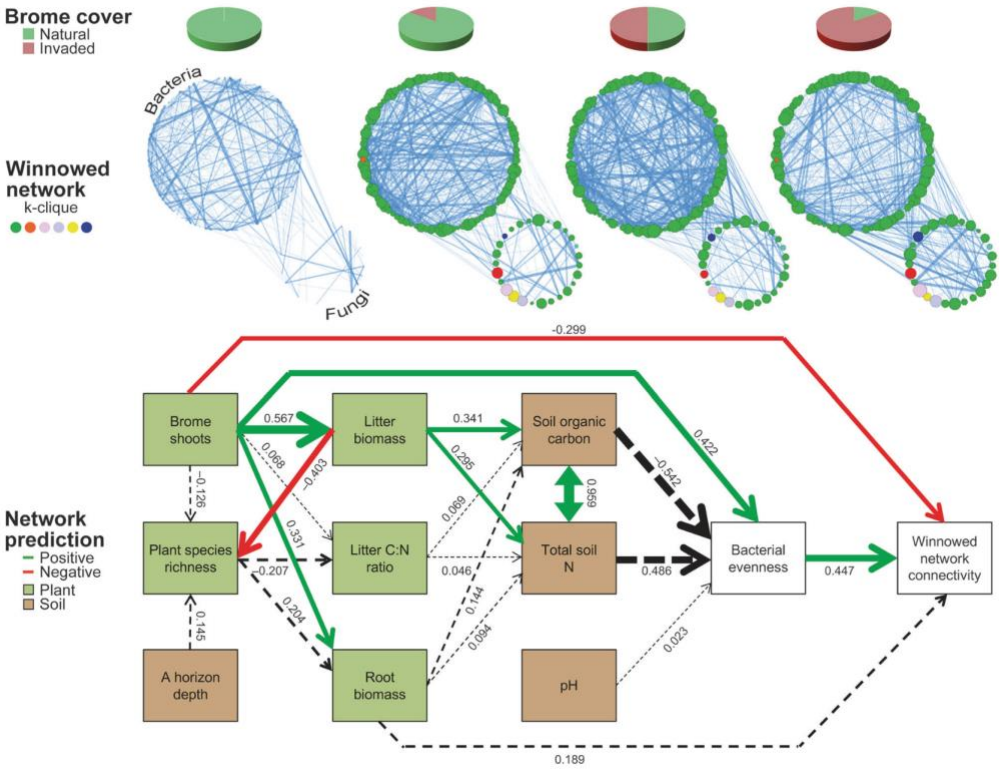

Figure 6.2 The resulting network prediction from Mamet et al.

6.3 Future Work

The current functionality implemented in the pipeline was chosen as a baseline to test the validity of the pipeline as a winnowing method for use in downstream analysis. As discussed above, the results of our pipeline were successfully used in this manner. This leads to the consideration of expanding the pipeline to implement further functionality. Fortunately, the modular architecture of the pipeline lends itself to easily incorporating future work.

Additional parameter options can be added to the existing pipeline. We chose four common graph centralities to support, but there are many more that could be added in the future, such as Katz or PageRank. Other simple parameters to expand on are the conditioning type and correlation type. As discussed in section 5.2, there were a number of instances where the eigenvector centrality did not return any or all results. This may be attributed to the implementation we chose to use. The default `networkx` function uses the power iteration method to determine the largest eigenvalue, to a maximum of 100 iterations. The pipeline could be updated to include a parameter to specify the maximum number of iterations. Though increasing the number of iterations may lead to the function returning results, the pipeline will take more time to complete due to the overhead of running more iterations.

The pipeline could also incorporate different modules completely. For example, as conceptualized in Figure 6.3 we could easily integrate alternate measures, to substitute for the network analysis piece.

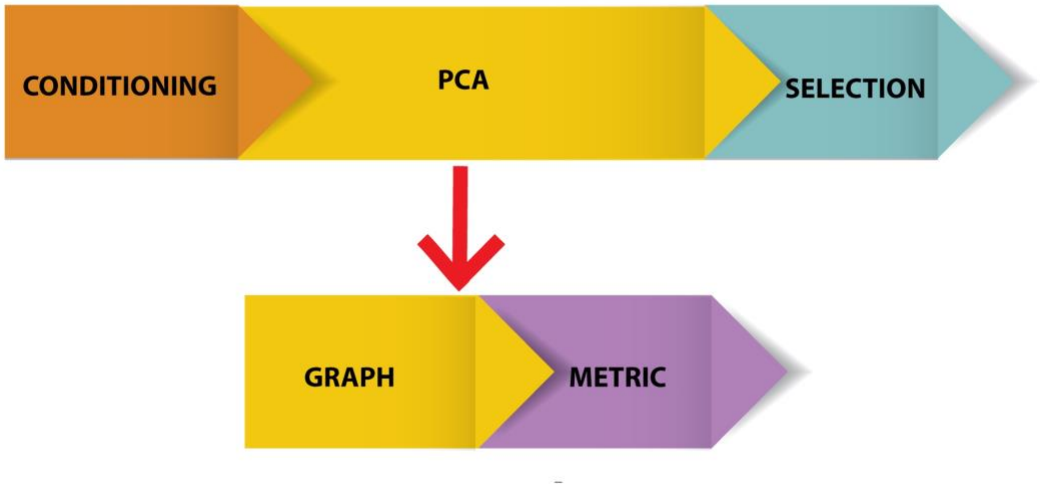

Figure 6.3 Example of exchanging pieces of the pipeline for new metrics.

There is an opportunity to add and improve the functionality of the sensitivity analysis. The sensitivity analysis and its evaluation could be added as final steps of the pipeline. Currently, these steps are separate from the pipeline and are used to analyze the results returned from the pipeline. We analyzed similarity and consistency in the results returned from the sensitivity analysis, but we did not use that analysis to determine a final, 'most stable' list of selected OTUs. An opportunity exists to incorporate this as another option in the pipeline.

6.4 Summary

This thesis proposes a feature selection pipeline using network analysis to aid in the analysis of high dimensional data sets. We were particularly interested in analyzing soil microbe data because the large number of organisms in the soil compared to the relatively few samples available has led to difficulties using traditional statistical techniques. This type of data is an excellent example of the "large *p*, small *n*" problem. We focused on using network analysis to select features that were identified as important based on different meanings of importance.

The pipeline was designed and implemented in a parameterized and modular fashion, which lends itself to flexibility in use and easy future development. To evaluate the impact that different combinations of parameters on the results, we performed a sensitivity analysis on a smooth brome invasion dataset. We assessed the sensitivity analysis with regards to reliability, performance, and result consistency. This allowed us to discuss the potential benefits and trade-

offs of the different parameters. We recognize that users may prioritize different aspects from our evaluation and highlighted some of the parameters that may be best suited for their needs.

Though we based our pipeline off network analysis, the modularity of the pipeline allows different types of analysis to be fit into the pipeline as an extension of this work. The current pipeline has already proven to be useful in downstream analysis. This work has given insight into how feature selection can be achieved on data sets with many features and few samples.

REFERENCES

[1] O. Tanaseichuk, J. Borneman, and T. Jiang, "Phylogeny-based classification of microbial communities," *Bioinformatics*, vol. 30, no. 4, pp. 449–456, 2014.

[2] G. W. Tyson *et al.*, "Community structure and metabolism through reconstruction of microbial genomes from the environment.," *Nature*, vol. 428, no. 6978, pp. 37–43, 2004.

[3] C. Lozupone *et al.*, "Identifying genomic and metabolic features that can underlie early successional and opportunistic lifestyles of human gut symbionts," *Genome Res.*, vol. 22, no. 10, pp. 1974–1984, 2012.

[4] S. Freilich, A. Kreimer, I. Meilijson, U. Gophna, R. Sharan, and E. Ruppin, "The large-scale organization of the bacterial network of ecological co-occurrence interactions," *Nucleic Acids Res.*, vol. 38, no. 12, pp. 3857–3868, 2010.

[5] K. Faust *et al.*, "Microbial Co-occurrence Relationships in the Human Microbiome," *PLOS Comput. Biol.*, vol. 8, no. 7, p. e1002606, Jul. 2012.

[6] A. Shade *et al.*, "Conditionally rare taxa disproportionately contribute to temporal changes in microbial diversity," *MBio*, vol. 5, no. 4, pp. e01371-14, Aug. 2014.

[7] S. Hacquard *et al.*, "Microbiota and host nutrition across plant and animal kingdoms," *Cell Host Microbe*, vol. 17, no. 5, pp. 603–616, 2015.

[8] National Institutes of Health, "NIH Human Microbiome Project defines normal bacterial makeup of the body," 2012. [Online]. Available: https://www.nih.gov/news-events/news-releases/nih-human-microbiome-project-defines-normal-bacterial-makeup-body.

[9] X. Raynaud and N. Nunan, "Spatial ecology of bacteria at the microscale in soil," *PLoS One*, vol. 9, no. 1, p. e87217, Jan. 2014.

[10] D. Tilman, C. Balzer, J. Hill, and B. L. Befort, "Global food demand and the sustainable intensification of agriculture," *Proc. Natl. Acad. Sci.*, vol. 108, no. 50, pp. 20260–20264, 2011.

[11] L. K. Wang;, V. Ivanov;, J.-H. Tay;, and Y.-T. Hung, "Microbial Ecology," in *Handbook of Environmental Engineering*, vol. 10, 2010, pp. 121–122.

[12] K. Faust and J. Raes, "Microbial interactions: From networks to models," *Nat. Rev. Microbiol.*, vol. 10, no. 8, pp. 538–550, 2012.

[13] T. Zhang, M. F. Shao, and L. Ye, "454 Pyrosequencing reveals bacterial diversity of

activated sludge from 14 sewage treatment plants," *ISME J.*, vol. 6, no. 6, pp. 1137–1147, 2012.

[14] P. Shi, A. Zhang, and H. Li, "Regression analysis for microbiome compositional data," *Ann. Appl. Stat.*, vol. 10, no. 2, pp. 1019–1040, 2016.

[15] X. Yan and J. Bien, "Rare Feature Selection in High Dimensions," *J. Am. Stat. Assoc.*, Mar. 2018.

[16] N. R. Pace, "A molecular view of microbial diversity and the biosphere," *Science (80-.).*, vol. 276, no. 5313, pp. 734–740, May 1997.

[17] T. S. B. Schmidt, J. F. Matias Rodrigues, and C. von Mering, "A family of interaction-adjusted indices of community similarity," *ISME J.*, vol. 11, no. 3, pp. 791–807, 2017.

[18] C. Ricotta and M. Marignani, "Computing β-Diversity with Rao's Quadratic Entropy: A Change of Perspective," *Divers. Distrib.*, vol. 13, no. 2, pp. 237–241, 2007.

[19] H. Yang *et al.*, "An integrated insight into the relationship between soil microbial community and tobacco bacterial wilt disease," *Front. Microbiol.*, vol. 8, no. NOV, pp. 1–11, 2017.

[20] Y. Deng, Y.-H. H. Jiang, Y. Yang, Z. He, F. Luo, and J. Zhou, "Molecular ecological network analyses," *BMC Bioinformatics*, vol. 13, no. 1, p. 113, 2012.

[21] B. Ma *et al.*, "Geographic patterns of co-occurrence network topological features for soil microbiota at continental scale in eastern China.," *ISME J.*, vol. 10, no. 8, pp. 1–11, 2016.

[22] I. M. Johnstone and D. M. Titterington, "Statistical challenges of high-dimensional data," *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 367, no. 1906, pp. 4237–4253, 2009.

[23] M. Lupatini *et al.*, "Network topology reveals high connectance levels and few key microbial genera within soils," *Front. Environ. Sci.*, vol. 2, no. MAY, pp. 1–11, 2014.

[24] R. J. Williams, A. Howe, and K. S. Hofmockel, "Demonstrating microbial co-occurrence pattern analyses within and between ecosystems," *Front. Microbiol.*, vol. 5, no. JULY, pp. 1–10, 2014.

[25] E. Corel, P. Lopez, R. Méheust, and E. Bapteste, "Network-Thinking: Graphs to Analyze Microbial Complexity and Evolution," *Trends Microbiol.*, vol. 24, no. 3, pp. 224–237, 2016.

[26] H. Wang *et al.*, "Combined use of network inference tools identifies ecologically meaningful bacterial associations in a paddy soil," *Soil Biol. Biochem.*, vol. 105, pp. 227–

235, 2017.

[27] S. Banerjee, C. A. Kirkby, D. Schmutter, A. Bissett, J. A. Kirkegaard, and A. E. Richardson, “Network analysis reveals functional redundancy and keystone taxa amongst bacterial and fungal communities during organic matter decomposition in an arable soil,” *Soil Biol. Biochem.*, vol. 97, pp. 188–198, 2016.

[28] S. D. Mamet, E. G. Lamb, C. L. Piper, T. Winsley, and S. D. Siciliano, “Archaea and bacteria mediate the effects of native species root loss on fungi during plant invasion,” *ISME J.*, vol. 11, no. 5, pp. 1261–1275, 2017.

[29] C. L. Piper, S. D. Siciliano, T. Winsley, and E. G. Lamb, “Smooth brome invasion increases rare soil bacterial species prevalence, bacterial species richness and evenness,” *J. Ecol.*, vol. 103, no. 2, pp. 386–396, 2015.

[30] S. D. Mamet *et al.*, “Structural equation modeling of a winnowed soil microbiome identi fi es how invasive plants re-structure microbial networks,” *ISME J.*, vol. 13, no. 8, pp. 1988–1996, 2019.

[31] T. Thomas, J. Gilbert, and F. Meyer, “Metagenomics - a guide from sampling to data analysis,” *Microb. Inform. Exp.*, vol. 2, no. 1, p. 3, 2012.

[32] P. D. Schloss, D. Gevers, and S. L. Westcott, “Reducing the effects of PCR amplification and sequencing artifacts on 16S rRNA-based studies.,” *PLoS One*, vol. 6, no. 12, p. e27310, Jan. 2011.

[33] C. R. Woese and G. E. Fox, “Phylogenetic structure of the prokaryotic domain: the primary kingdoms.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 74, no. 11, pp. 5088–5090, 1977.

[34] M. D. J. Lynch, J. D. Neufeld, M. D. J Lynch, and J. D. Neufeld, “Ecology and exploration of the rare biosphere,” *Nat. Publ. Gr.*, vol. 13, no. 4, pp. 217–229, 2015.

[35] B. J. Haas *et al.*, “Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons,” *Genome Res.*, vol. 21, no. 3, pp. 494–504, 2011.

[36] J. E. Hill, S. L. Penny, K. G. Crowell, S. H. Goh, and S. M. Hemmingsen, “cpnDB : A Chaperonin Sequence Database cpnDB : A Chaperonin Sequence Database,” *Genome Res.*, vol. 14, no. 8, pp. 1669–1675, 2004.

[37] J. Schellenberg *et al.*, “Pyrosequencing of the chaperonin-60 universal target as a tool for determining microbial community composition,” *Appl. Environ. Microbiol.*, vol. 75, no. 9, pp. 2889–2898, 2009.

[38] S. M. Hemmingsen *et al.*, “Homologous plant and bacterial proteins chaperone oligomeric protein assembly,” *Nature*, vol. 333, no. 6171, pp. 330–334, 1988.

[39] H. Teeling and F. O. Glockner, “Current opportunities and challenges in microbial metagenome analysis-A bioinformatic perspective,” *Brief. Bioinform.*, vol. 13, no. 6, pp. 728–742, 2012.

[40] R. Poretsky, L. M. Rodriguez-R, C. Luo, D. Tsementzi, and K. T. Konstantinidis, “Strengths and limitations of 16S rRNA gene amplicon sequencing in revealing temporal microbial community dynamics.,” *PLoS One*, vol. 9, no. 4, p. e93827, Jan. 2014.

[41] W. Koh *et al.*, “Analysis of deep sequencing microRNA expression profile from human embryonic stem cells derived mesenchymal stem cells reveals possible role of let-7 microRNA family in downstream targeting of hepatic nuclear factor 4 alpha,” *BMC Genomics*, vol. 11, no. SUPPL. 1, 2010.

[42] R. Daniel, “The metagenomics of soil,” *Nat. Rev. Microbiol.*, vol. 3, no. 6, pp. 470–478, 2005.

[43] B. K. Singh, C. D. Campbell, S. J. Sorenson, and J. Zhou, “Soil genomics,” *Nat. Rev. Microbiol.*, vol. 7, no. 10, p. 756, 2009.

[44] M. S. Elshahed *et al.*, “Novelty and uniqueness patterns of rare members of the soil biosphere,” *Appl. Environ. Microbiol.*, vol. 74, no. 17, pp. 5422–5428, 2008.

[45] A. Reid, M. Buckley, and M. Mcfall-, “The Rare Biosphere,” *Am. Acad. Microbiol.*, 2011.

[46] D. B. M. Welch and S. M. Huse, “Microbial Diversity in the Deep Sea and the Underexplored ‘Rare Biosphere,’” *Handb. Mol. Microb. Ecol. II Metagenomics Differ. Habitats*, no. 30, pp. 243–252, 2011.

[47] Y. Cao, D. D. Williams, and N. E. Williams, “How important are rare species in aquatic community ecology and bioassessment?,” *Limnol. Oceanogr.*, vol. 43, no. 7, pp. 1403–1409, Nov. 1998.

[48] Q. Guo, J. H. Brown, and T. J. Valone, “Abundance and distribution of desert annuals: Are spatial and temporal patterns related?,” *J. Ecol.*, vol. 88, no. 4, pp. 551–560, 2000.

[49] R. Winfree, J. W. Fox, N. M. Williams, J. R. Reilly, and D. P. Cariveau, “Abundance of common species, not species richness, drives delivery of a real-world ecosystem service,” *Ecol. Lett.*, vol. 18, no. 7, pp. 626–635, 2015.

[50] S. Banerjee, K. Schlaeppi, and M. G. A. van der Heijden, “Keystone taxa as drivers of

microbiome structure and functioning," *Nat. Rev. Microbiol.*, vol. 16, no. 9, pp. 567–576, 2018.

[51] A. Jousset *et al.*, "Where less may be more: How the rare biosphere pulls ecosystems strings," *ISME J.*, vol. 11, no. 4, pp. 853–862, 2017.

[52] M. Pester, N. Bittner, P. Deevong, M. Wagner, and A. Loy, "A 'rare biosphere' microorganism contributes to sulfate reduction in a peatland," *ISME J.*, vol. 4, no. 12, pp. 1–12, 2010.

[53] A. Shade and J. Handelsman, "Beyond the Venn diagram: The hunt for a core microbiome," *Environ. Microbiol.*, vol. 14, no. 1, pp. 4–12, 2012.

[54] S. E. Jones and J. T. Lennon, "Dormancy contributes to the maintenance of microbial diversity," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 107, no. 13, pp. 5881–5886, 2010.

[55] J. Chen, F. D. Bushman, J. D. Lewis, G. D. Wu, and H. Li, "Structure-constrained sparse canonical correlation analysis with an application to microbiome data analysis," *Biostatistics*, vol. 14, no. 2, pp. 244–258, 2013.

[56] Y. Cao, A. Zhang, and H. Li, "Multi-sample Estimation of Bacterial Composition Matrix in Metagenomics Data," *arXiv Methodol.*, no. June, 2017.

[57] S. Weiss *et al.*, "Correlation detection strategies in microbial data sets vary widely in sensitivity and precision," *ISME J.*, vol. 10, no. 7, pp. 1669–1681, 2016.

[58] Q. Ruan, D. Dutta, M. S. Schwalbach, J. A. Steele, J. A. Fuhrman, and F. Sun, "Local similarity analysis reveals unique associations among marine bacterioplankton species and environmental factors," *Bioinformatics*, vol. 22, no. 20, pp. 2532–2538, 2006.

[59] D. N. Reshef *et al.*, "Detecting Novel Associations in Large Datasets," *Science (80-.).*, vol. 334, no. 6062, pp. 1518–1524, Dec. 2012.

[60] F. Luo *et al.*, "Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory," *BMC Bioinformatics*, vol. 8, pp. 1–17, 2007.

[61] F. Luo, J. Zhong, Y. Yang, R. H. Scheuermann, and J. Zhou, "Application of random matrix theory to biological networks," *Phys. Lett. Sect. A Gen. At. Solid State Phys.*, vol. 357, no. 6, pp. 420–423, 2006.

[62] K. Faust and J. Raes, "CoNet app: Inference of biological association networks using Cytoscape [version 1; referees: 2 approved with reservations]," *F1000Research*, vol. 5, pp. 1–16, 2016.

[63] J. M. Beman, J. A. Steele, and J. A. Fuhrman, “Co-occurrence patterns for abundant marine archaeal and bacterial lineages in the deep chlorophyll maximum of coastal California,” *ISME J.*, vol. 5, no. 7, pp. 1077–1085, 2011.

[64] K. C. Li, “Genome-wide coexpression dynamics: Theory and application,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 99, no. 26, pp. 16875–16880, 2002.

[65] L. C. Xia, D. Ai, J. Cram, J. A. Fuhrman, and F. Sun, “Efficient statistical significance approximation for local similarity analysis of high-throughput time series data,” *Bioinformatics*, vol. 29, no. 2, pp. 230–237, 2013.

[66] J. L. Gross and J. Yellen, *Handbook of Graph Theory, Second Edition*, vol. 20134658. 2013.

[67] W. D. Joyner, *Adventures in Graph Ramsey Theory*, 1st ed. Birkhäuser Basel, 2017.

[68] G. A. Pavlopoulos *et al.*, “Using graph theory to analyze biological networks.,” *BioData Min.*, vol. 4, p. 10, 2011.

[69] A. Barberan, S. T. Bates, E. O. Casamayor, and N. Fierer, “Using network analysis to explore co-occurrence patterns in soil microbial communities,” *ISME J.*, vol. 6, no. 2, pp. 343–351, 2012.

[70] M. Layeghifard, D. M. Hwang, and D. S. Guttman, “Disentangling Interactions in the Microbiome: A Network Perspective,” *Trends Microbiol.*, vol. 25, no. 3, pp. 217–228, 2017.

[71] “Spearman Rank Correlation Coefficient,” in *The Concise Encyclopedia of Statistics*, New York, NY: Springer New York, 2008, pp. 502–505.

[72] E. Zotenko, J. Mestre, D. P. O’Leary, and T. M. Przytycka, “Why do hubs in the yeast protein interaction network tend to be essential: Reexamining the connection between the network topology and essentiality,” *PLoS Comput. Biol.*, vol. 4, no. 8, 2008.

[73] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai, “Lethality and centrality in protein networks,” *Nature*, vol. 411, no. 6833, pp. 41–42, 2001.

[74] H. W. Ma and A. P. Zeng, “The connectivity structure, giant strong component and centrality of metabolic networks,” *Bioinformatics*, vol. 19, no. 11, pp. 1423–1430, 2003.

[75] M. R. Da Silva, H. Ma, and A. P. Zeng, “Centrality, network capacity, and modularity as parameters to analyze the core-periphery structure in metabolic networks,” *Proc. IEEE*, vol. 96, no. 8, pp. 1411–1420, 2008.

[76] F. Boudin and L. U. M. R. Cnrs, “A Comparison of Centrality Measures for Graph-Based Keyphrase Extraction,” no. October, pp. 834–838, 2013.

[77] A. L. Barabási, N. Gulbahce, and J. Loscalzo, “Network medicine: A network-based approach to human disease,” *Nat. Rev. Genet.*, vol. 12, no. 1, pp. 56–68, 2011.

[78] M. E. J. Newman, “The structure and function of complex networks,” *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.

[79] K. Bryan and T. Leise, “The Linear Algebra behind Google ∗,” vol. 48, no. 3, pp. 569–581, 2006.

[80] A. Özgür, T. Vu, G. Erkan, and D. R. Radev, “Identifying gene-disease associations using centrality on a literature mined gene-interaction network,” *Bioinformatics*, vol. 24, no. 13, pp. 277–285, 2008.

[81] D. Antonenko *et al.*, “Age-dependent effects of brain stimulation on network centrality,” *Neuroimage*, vol. 176, no. November 2017, pp. 71–82, 2018.

[82] W. Webber, A. Moffat, and J. Zobel, “A similarity measure for indefinite rankings,” *ACM Trans. Inf. Syst.*, vol. 28, no. 4, pp. 1–38, 2010.

[83] L. Tan and C. L. A. Clarke, “A Family of Rank Similarity Measures Based on Maximized Effectiveness Difference,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 2865–2877, 2015.

[84] C. Rao, “A review of canonical coordinates and an alternative to correspondence analysis using Hellinger distance,” *Questiio*, vol. 19, no. 1–3. pp. 23–63, 1995.

[85] P. Legendre and E. D. Gallagher, “Ecologically meaningful transformations for ordination of species data,” *Oecologia*, vol. 129, no. 2, pp. 271–280, 2001.

[86] A. Ramette, “Multivariate analyses in microbial ecology,” *FEMS Microbiol. Ecol.*, vol. 62, no. 2, pp. 142–160, 2007.

[87] C. Tebby, S. Joachim, P. J. Van den Brink, J. M. Porcher, and R. Beaudouin, “Analysis of community-level mesocosm data based on ecologically meaningful dissimilarity measures and data transformation,” *Environ. Toxicol. Chem.*, vol. 36, no. 6, pp. 1667–1679, 2017.

[88] M. Clark, “A Comparison Of Correlation Measures,” *Nd.Edu*, 2013.

[89] J. G. Caporaso *et al.*, “Global patterns of 16S rRNA diversity at a depth of millions of sequences per sample,” *Proc. Natl. Acad. Sci.*, vol. 108, no. Supplement_1, pp. 4516–4522, 2011.

[90] P. D. Schloss *et al.*, “Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities,” *Appl. Environ. Microbiol.*, vol. 75, no. 23, pp. 7537–7541, 2009.

[91] K. G. Frey *et al.*, “Comparison of three next-generation sequencing platforms for metagenomic sequencing and identification of pathogens in blood,” *BMC Genomics*, vol. 15, no. 1, pp. 1–14, 2014.

[92] A. Gobet, C. Quince, and A. Ramette, “Multivariate Cutoff Level Analysis (MultiCoLA) of large community data sets,” *Nucleic Acids Res.*, vol. 38, no. 15, p. e155, 2010.

[93] A. Zhan, W. Xiong, S. He, and H. J. MacIsaac, “Influence of artifact removal on rare species recovery in natural complex communities using high-throughput sequencing,” *PLoS One*, vol. 9, no. 5, 2014.

APPENDIX A

The feature selection pipeline is highly parameterizable. For the purposes of our sensitivity analysis, the parameters described below were utilized:

- Input file – the filename of the abundance dataset in csv format that will be analyzed.
- metric – the metric to use. This is graph centrality for our analyses, but another metric, such as PCA, could be implemented
- minimum count – the minimum total abundance count of a feature to be considered in the analysis. Any features with this minimum number or fewer will be removed from the dataset before analysis.
- conditioning – the type of data transformation to perform on the data prior to analysis.
- Select total – the total number of features to select in analysis.
- Select per iteration – the number of features that should be selected at each iteration through the pipeline without replacement. For example, if the user wants to select 50 features total and select 5 per iteration, the pipeline process will be run 10 times. For the first iteration of the pipeline, the full dataset will be used to condition and build the graph. The top 5 selected features will be selected and removed from the dataset the process will be run again. This will continue until the 10^{th} iteration is complete and all 50 features have been selected. In some cases, when selecting the 50^{th} feature, there are multiple features with the highest remaining centrality. Because there is no justifiable way to determine the most important of these, all tied features will be included in the result set. The rationale behind this type of selection process is to discern if there is a difference in the results when important nodes are removed and the graphs are recreated. More specifically, we look to answer: if we remove a main hub from a graph and build a new graph with the remaining features, will the network structure be changed such that the resulting selected nodes would be different than if they had been selected from the initial graph?
- Centrality – when using the graph metric, this is type of graph centrality to calculate the 'importance' of the features. Currently, betweenness, degree, closeness, and eigenvector can be used for the centrality options.
- Correlation type – when using the graph metric, this is the type of correlation to use to build the network.

- Threshold – when using the graph metric, this is the threshold value to remove weak edges. At each iteration, after the network is created using the specified correlation, any edges with correlations less than the threshold will be removed from the network.
- Weight – When using the graph metric, the weighting is a Boolean parameter that specifies if the network edges should have weights assigned to them. If weighting is used, the centrality will consider the weight value when calculating the centrality values.
- Correlation property – When using the graph metric, this parameter specifies if the network should consider either positive or negative correlations, or both. We used both positive and negative correlations in our analysis because we are interested biologically in things that move together.
- Percent connected - When using the graph metric, this parameter specifies if the graph should be evaluated if the largest connected subgraph doesn't make up a certain percentage of the entire network. The metric step will only continue if the largest connected subgraph makes up the percentage value specified or higher. This is a positive integer value between 0 and 100.